July 2023

Volume 14 Issue 1

JoCsE

Journal Of Computational Science Education

Promoting the Use of Computational Science Through Education

ISSN 2153-4136 (online)

Volume 14, Issue 1 **July 2023**

JOCSE

Journal Of Computational Science Education

The Journal of Computational Science Education (JOCSE), ISSN 2153-4136, published in online form, is a supported publication of the Shodor Education Foundation Incorporated. Materials accepted by JOCSE will be hosted on the JOCSE website and will be catalogued by the Computational Science Education Reference Desk (CSERD) for inclusion in the National Science Digital Library (NSDL).

Subscription: JOCSE is a freely available online peer-reviewed publication which can be accessed at `http://jocse.org`.

CONTENTS

Introduction to Volume 14, Issue 1

David Joiner
Editor
Kean University
Union, NJ
djoiner@kean.edu

FOREWORD

In this issue, we present papers from the SC22 Ninth Workshop on Best Practices for HPC Education and Training, and two additional papers.

Horn and colleagues put forth an innovative pedagogical strategy that enhances the teaching of network security within the realm of computer engineering. They have developed a unique curricular approach that focuses on protocol behavior and trust point observations, creating a novel path towards understanding and learning secure design of networks.

In their paper, Lu and Lampert emphasize the significant role Python can play in environmental modeling. They showcase how Python can be used to simulate the movement of substances within porous media, with examples of how this can be used for student engagement.

Barker and colleagues address the challenge of expanding the HPC workforce, emphasizing the role of hybrid and virtual hackathons in bridging the gap between traditional programming and necessary hands-on skills. They provide an overview of current programs, insights from past hackathons, and offer implementation recommendations.

Gyires-Tóth et. al. discuss the importance of accelerated computing and deep learning, acknowledging the unique expertise needed in these fields. They explore the teaching methodology of the NVIDIA Deep Learning Institute, present post-workshop survey results, and provide a case study on teaching heterogeneous parallel computing.

Lee, a student researcher, introduces a novel approach to genetic sequencing and bioinformatics using quantum annealers. The paper presents a modified MSA algorithm that leverages the properties of quantum mechanics to overcome the computational challenges of aligning extensive sets of genetic sequences. While traditional algorithms rely on brute force or heuristic methods, this new approach uses progressive alignment techniques to optimize quantum annealing algorithms.

Mensa et al. focus on training users in hybrid technologies integrated with high-performance computing (HPC). They propose a three-stage education plan, which involves foundational HPC training, digital innovation awareness, and specialized training tailored to business needs. The approach aims to enhance productivity and encourage the adoption of innovative practices.

Ngo and Bui address the difficulties inherent in big data education and propose a comprehensive solution. Their paper suggests a dual approach that leverages both personal computers and public cloud resources to provide meaningful, hands-on learning experiences, helping students gain practical expertise in big data analysis.

Parete-Koon and colleagues offer an overview of the U.S. Department of Energy's Exascale Computing Project's initiative to diversify the HPC workforce. Their work highlights efforts to create a sustainable and inclusive culture within the computing sciences, with the goal of attracting and retaining a diverse group of professionals.

Lastly, Biggerstaff et al. provide a compelling demonstration of how computational tools can be applied to tackle a global health crisis. Their research focuses on identifying potential inhibitors for the SARS-CoV-2 virus, showcasing the critical role of computational analysis in advancing antiviral drug discovery.

As I embark on my journey as the new editor of JOCSE, I'd like to express my deep appreciation for the formidable legacy left by our founding editor, Steve Gordon. His relentless commitment to cultivating and elevating this journal has set a high bar for those who follow. I also owe a significant debt of gratitude to Aaron Weeden, whose technical acumen has shaped the face of JOCSE in the past years. His invaluable work on consolidating past issues and enhancing our back-end infrastructure over the past year has left an indelible mark. Lastly, but by no means least, my heartfelt thanks to Holly Hirst for adeptly stepping into Aaron's role in assembling and circulating this issue. I look forward to seeing the contributions we will make together to JOCSE in the years to come.

Representing Patterns of Learning as a Function of Course Opportunities

Catherine Horn
University of Houston
Houston, TX
clhorn2@uh.edu

Deniz Gurkan
University of Houston
Houston, TX
dgurkan@uh.edu

Jennifer Chauvot
University of Houston
Houston, TX
jchauvot@central.uh.edu

ABSTRACT

Expanded articulation of demonstrable competencies and a burgeoning demand for security analysts increasingly responsive to rapidly evolving conditions have brought to foreground a need to revamp core curriculum in the area. Once such effort has emerged at one university where a faculty member in computer engineering technology, network communications, and computer science has developed a novel pedagogical strategy that teaches network security through protocol behavior and trust point observations. This paper used a single course case study to explore the engagement patterns of learning associated with this novel curricular approach to learning secure design of networks. This exploratory study's findings lay important foundation for understanding the ways in which students are making use of multiple forms of experiential engagement. While homework exercises, perhaps conceptually the most traditional form of engagement, were accessed largely at a one opportunity per student count, practices and much more importantly labs were used in much more frequent ways. In particular, labs display a positive engagement patten in that they demonstrate students' choices to access early and in a sustained variety of topics. Importantly, these opportunities are active in their mechanism for learning, which connects with a strategy previous empirical literature has positively reinforced.

KEYWORDS

Interactive Learning, Computer Science Education, Scaffolded Learning, Computer Networks

1 INTRODUCTION

Expanded articulation of demonstrable competencies and a burgeoning demand for security analysts who are increasingly responsive to rapidly evolving conditions have brought to foreground a need to revamp core curriculum in the area. Specifically, federal and agency guidelines prompt instructors to consider differently their approach to cybersecurity education in order to better prepare graduates [4, 10, 13, 15, 16].

Once such effort has emerged at the University of Houston where a faculty member in computer engineering technology, network communications, and computer science has developed a novel pedagogical strategy that teaches network security through protocol behavior and trust point observations [9]. Specifically, this undergraduate class is designed to introduce the concept of trust protocol points and guiding principles through a scaffolded set of learning opportunities available to students in a semi-autonomous opportunity to learn. The course combines lectures, hands-on labs, homework, auto-graded practices, exams, and a final project to allow multiple opportunities for students to master material (see [9] for full description).

This paper uses a single course case study to explore the engagement patterns of learning associated with this novel curricular approach to learning secure design of networks. Specifically, the study seeks to answer the following research questions: What patterns of learning engagement do students demonstrate? What pedagogical tools associate with these patterns?

2 THE ROLE OF EXPERIENTIAL LEARNING OPPORTUNITIES

Abundant literature documents the definitions and benefits of experiential learning in knowledge development [3, 8, 11, 14, 17]. As a generic representation, Table 1 presents a scale of learning typologies. As it indicates, learning experiences move from more concrete to abstract where experiences also vary from more active to ones where students function largely as receivers of information. As Bersteinger et al. summarize, "primary learning essentially occurs through active/concrete doing, whereas secondary learning occurs when a passive receiver interprets abstract information communicated by another through spoken words, written text, graphic images or gestures" [2, p. 37].

Table 1. Generic Scale of learning typologies [13].

Concrete/Active		Abstract/Passive	
Student as actor		Student as receiver	
Do an activity	Watch an activity	Hear about an activity	Read about an activity

In a related literature, research has identified the utility of teaching and learning through multiple strategies toward student learning. In his seminal work, Howard Gardner [7] posited a theory of multiple intelligences where learners differ in their capacity and preference for different forms of information processing based on the kinds of intelligence they demonstrate. Taken together, these bodies of research suggest the need for varied learning opportunities where at least part of those experiences ground in experiential learning.

2.1 Conceptual frames guiding engagement through experiences

Complex Adaptive Systems Theory (CAS) guides this study's understanding of engagement patterns and the instructional tools that associate with them (Figure 1). First, CAS suggests that knowledge develops through novel encounters with information and other opportunities to ultimately formulate a set of rules to guide understanding. Key to this process is the role of feedback where knowledge developers have opportunity to adapt their

DOI: https://doi.org/10.22369/issn.2153-4136/14/1/1

understanding based on attempts and useful response to those attempts. At its core, CAS assumes the presence of an organizational structure that shapes and is shaped by knowledge formation. In the case of this study, the classroom itself serves as the "human organization" where the students are encouraged to "innovate by producing spontaneous, systemic bouts of novelty out of which new patterns of behavior emerge. Patterns which enhance a system's ability to adapt successfully to its environment are stabilized and repeated; those that do not are rejected in favor of radically new ones, almost as if a cosmic game of trial-and-error were being played. Complexity is, therefore, in part, the study of pervasive innovation in the universe" [12, p. 196].

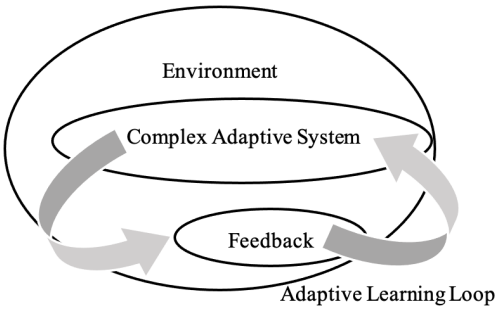


Figure 1. Complex Adaptive System (CAS) Model [16].

At the individual level, then, learning "is a process of emergence and co-evolution of the individual, the social group, and the wider society. Emphasis is placed on the relationship between elements, rather than the elements themselves" [12]. Through this frame, then, the current study seeks, then, to understand how engagement represents an evolution of novel to more sophisticated encounters with new information.

3 DATA AND METHODS

3.1 Participants and Learning Experiences

Data for this study were drawn from 58 students taking a 16-week undergraduate introduction to networking course offered in Fall 2020. Students completed 9 homework exercises throughout the duration of the course and received a full completion grade for any complete first attempt (regardless of correctness of answers). Because the purpose of the assignment was to serve as a developmental opportunity for learners to assess their understanding and work toward mastery in a low-stakes format, they were also provided feedback on the accuracy of each of their responses. Subsequently, students were permitted to return to any items they answered incorrectly and attempt them again (with new randomly generated data). The primary goal of the homework–in format and in function–was to provide an opportunity to strengthen their capacity to do well on the laboratory assignments, exam, and ultimately the assigned project.

Students also engaged in labs and practice over the course of the study. Labs, which are not graded, are opportunities intended to aid in homework submission. A lab manual web page provides detailed instructions and provide another opportunity through a different format to continue to engage in work toward mastery of a set of discrete but scaffolded concepts leading toward a comprehensive understanding of network security. Finally, practice opportunities are directly linked to discrete learning outcomes assessed in the homework and provide yet another space and structure for students to grapple with what they understand and what remains unclear with respect to specific competencies they are expected to develop.

Data for this study derive from the usage patterns for each of these opportunities, including assignment and date accessed.

3.2 Analytical Approach

This study descriptively represents patterns of engagement. Specifically, it aggregates the number of times a particular learning opportunity was accessed in total and by month. For labs and practices, Chi Square statistics were calculated to assess differences in distribution by opportunity and by month.

4 FINDINGS

Findings are organized around key aspects of the course: homework; labs; and practice opportunities.

4.1 Homework Exercises

Homework exercises were accessed a total of 534 times throughout the semester (M=59, SD=8.16). Figure 2 presents the distribution of engagement counts by homework exercise topic.

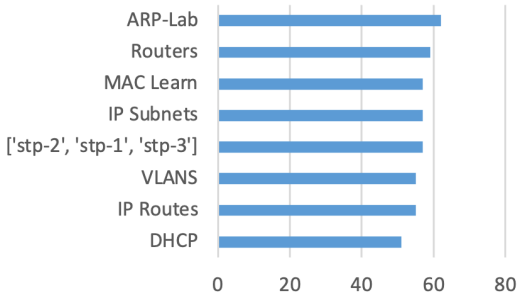


Figure 2. Fall 2020 ELET 4421 Exercises - Engagement Counts.

As can be seen, most exercises were accessed a similar number of times (on average, approximately 1 time per student in the class). In seeking to understand the extent to which students engaged and reengaged with homework exercises over time, Figure 3 in presents counts by week.

Two important observations are noted. First, not surprisingly, students engaged in most substantial numbers nearest the time when the exercise was on the syllabus related to topic of discussion. That said, for most exercises, distribution of access occurred over at least a 2- and sometimes a 3-week period. This is an important observation in that it suggests a fluidity of engagement among students with respect to learning opportunities. Second, as evidenced by the inclusion of "redisplaying current exercise module," execution of that command occurred more often during the initial weeks of the class. The gradual decline in redisplay suggests that as they learned to navigate the system, the need for re-execution waned.

4.2 Labs

Patterns of engagement in lab opportunities, in contrast to homework exercises, identified that this learning strategy was far more accessed overall and varied in relationship to particular units. Overall, labs were accessed 1881 times, an average of 32.4 times per student in the class. The mean number of times accessed per lab unit was 125 (SD=111.41). The next several subsections (4.2.1-4.2.4) briefly describe key labs before the paper turns to findings.

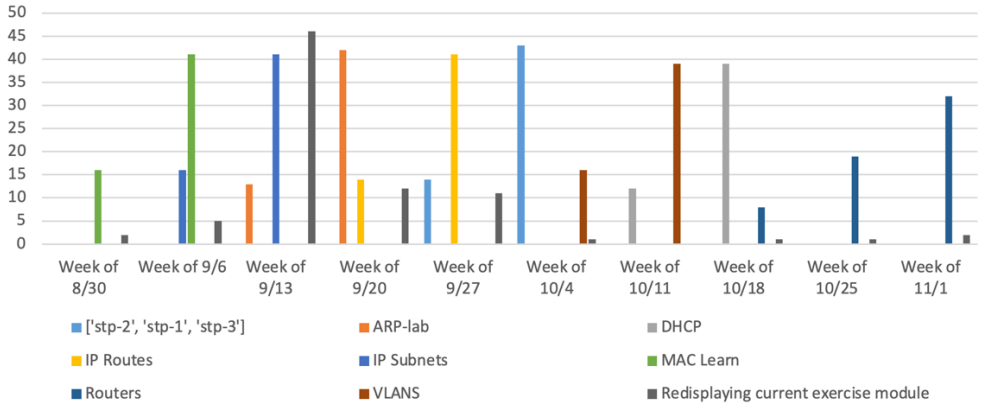


Figure 3. Exercise Engagement Counts by Week.

4.2.1Layer 2: Ethernet Lab

Broadcast domain concepts, layer 2 forwarding and MAC address learning functionality in Ethernet networks is covered through the four lab modules: Ethernet bridge MAC learning, ARP, VLANs and a host connected to a bridge (Figure 4). The observations are composed of sending and monitoring of packets on host interfaces, examination of bridge layer 2 tables, and bridge port configurations. Students are able to conduct experiments on network topologies that in turn allow them to verify the knowledge they are gaining while also new experiences are provided in network state observations, troubleshooting, and analysis of network topologies and protocols through packet traces and protocol behavior.

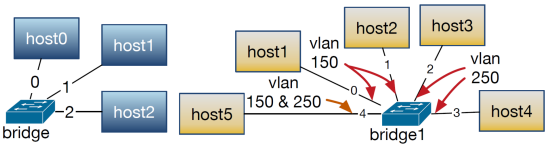


Figure 4. Ethernet bridges provide layer 2 connectivity and when port VLANs are configured, layer 2 isolation. The representative topologies that are used in the lab modules are included here.

4.2.2 IP Subnetting and Routing, Address Resolution Protocol

IP subnet assignment and bit math are introduced with example topologies and exercises that emphasize the calculations of IP subnets and host addresses within a subnet (Figure 5).

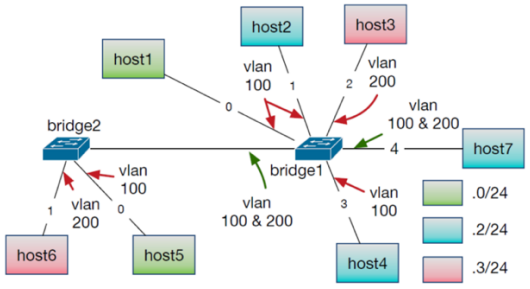


Figure 5. IP subnets are assigned inside two broadcast domains to hosts that are also isolated in the layer 2 broadcast domain.

Routing is introduced in representative topologies illustrated in the Figure 6 with router devices that forward between subnets as well as subnet addresses assigned to hosts with route tables that reflect the network state and configuration for reachability.

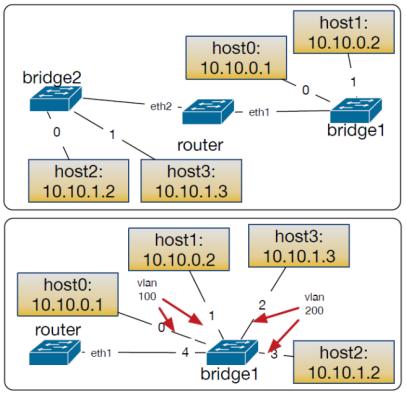


Figure 6. Routing and routers are utilized in the lab modules that cover ARP, IP routing, and route tables.

4.2.3 DNS and DHCP

Typical services that run in a network are DHCP and DNS. The services are included in the representative topologies shown in Figure 7.

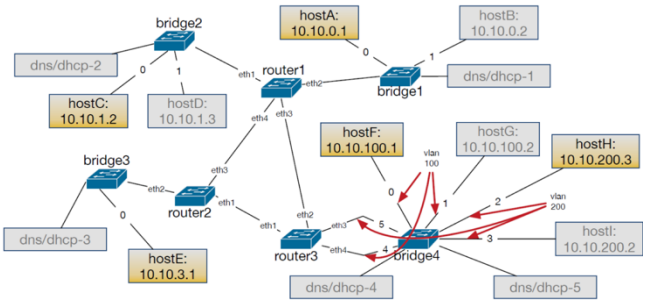


Figure 7. A number of subnets along with naming services are instantiated in network topologies.

The host interfaces are configured using the services in the network. Sample name resolutions are achieved to gain experience and firsthand understanding of the function and innerworkings of DNS protocol in the network.

4.2.4 Projects: Network Troubleshooting

The course final activity is culminating project where critical thinking is required to complete. Students are presented with networks that have misconfigurations. They are asked then to test reachability to identify the misconfigurations in the network devices and end hosts (See Figure 8). In the process, they are required to use the lab investigation methods they learned throughout the semester during the lab activities. They apply their knowledge of how network devices behave and what protocol observations they need to make to identify the misconfigurations. The students are also provided with the vendor-agnostic methods to correct the misconfigurations on their individual networks. The second phase of the project activity requires that the misconfigurations are corrected and full reachability is achieved in the submission phase.

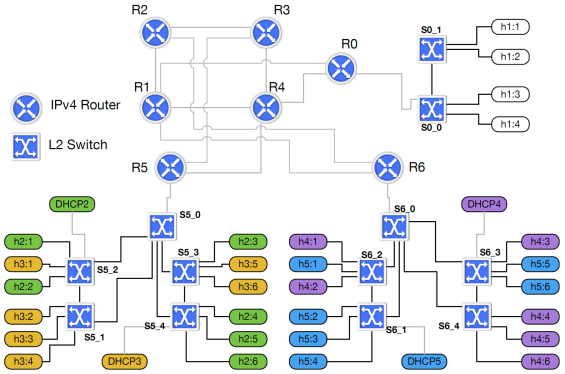


Figure 8. A topology that is pre-configured with typical misconfigurations is provided during the lab in order to teach network troubleshooting skills and reflect on the learnings in the previous labs.

4.2.5 Engagement Data

As reinforced in Figure 9, lab engagement ranged from 10 (STP Typology C) to 474 access records (Mismatch Typology Found; not displayed in Figure but happens when a student has forgotten to delete their existing topology from a previous lab and tries to build the next lab). The majority of labs were accessed between 72 and 141 times (an average of 1.24 and 2.43 times per student).

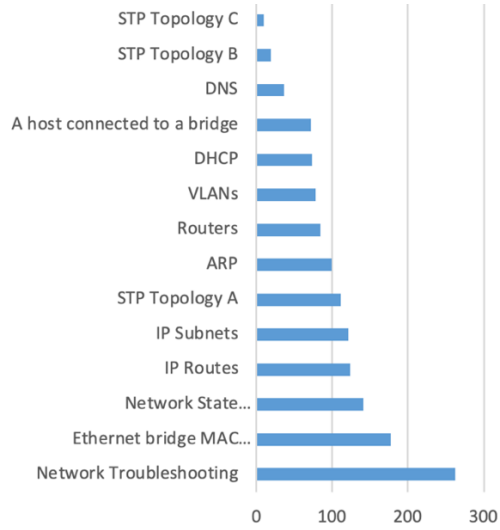


Figure 9. Fall 2020 ELET 4421 Labs – Engagement Counts.

When again considering patterns of engagement over time, Figure 10 identifies students engaged almost half (7 of 15) of the labs in the first month of the semester.

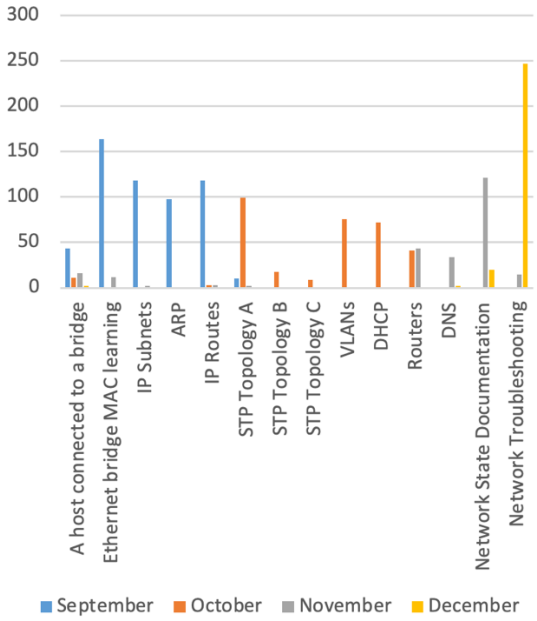


Figure 10. Fall 2020 ELET Labs – Engagement Counts by Month

Similarly, 8 of the labs were accessed in October. Students engaged with fewer labs (6) in October, and only 3 labs were accessed in December. This difference in total access counts across months is statistically significant ($\chi^2(3) = 650.36$, $p < .001$) as is the difference across months by specific lab ($\chi^2(42) = 3227.72$, $p < .001$).

4.3 Practice

Practice opportunities were engaged 949 times throughout the course of the semester with an average count of 38 encounters per discrete practice (SD=12.13). Similar to homework exercises, most practice opportunities, looking across the broader topical areas, were accessed between 98 and 121 times (between 1.69 and 2.09 times per student). Figure 11 presents the distribution across aggregated topical areas, indicating Loading ARP Practice as most accessed (189 times).

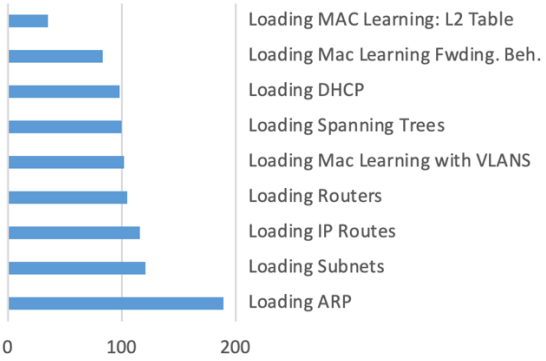


Figure 11. Fall 2020 ELET 4421 Practice Counts Aggregated by Major Topic Area.

Figure 12 presents access counts by month and identifies that all practice areas were exclusively or almost entirely accessed in November. Students are able to practice as soon as they submit a homework, which makes this finding especially important.

While practice on a topic becomes available immediately after homework is submitted, during the week leading up to the exam (in mid-November), all the items are made available. As such, it is important to understand weekly access patterns for practice during the concentrated month of engagement. Figure 8 displays counts, by week, for the month of November.

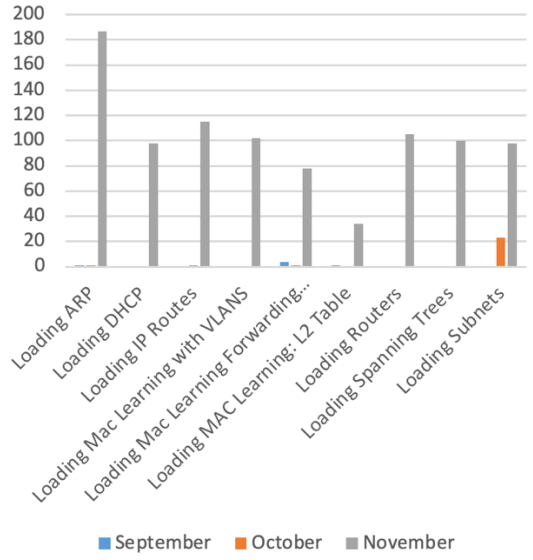


Figure 12. Practice Aggregated by Major Heading and by Month.

As can be seen, all the practice opportunities were accessed predominantly within a single week (week of November 7). Similarly, a majority are being visited or revisited (but at a lower count relative to the week of November 7) during the week of October 31.

5 DISCUSSION, IMPLICATIONS FOR PRACTICE, AND FUTURE RESEARCH

5.1 Discussion and Limitations

This exploratory study's findings lay important foundation for understanding the ways in which students are making use of multiple forms of experiential engagement. While homework exercises, perhaps conceptually the most traditional form of engagement are accessed largely at a one opportunity per student count, practices and much more importantly labs are used in much more frequent ways. In particular, labs display a positive engagement patten in that they demonstrate students' choices to access early and in a sustained a variety of topics. Importantly, these opportunities are active in their mechanism for learning, which connects with a strategy previous empirical research has positively reinforced.

The findings related to the ways students are engaging in practice is also an important one. In connection with the ways in which a complex adaptive system works, students are taking feedback (provided through original homework) to seek additional opportunities to refine understanding. Practices are equipped with an auto grader (correct/incorrect) that gives immediate feedback when utilized. However, the findings of this study suggest that rather than associating that extended learning more proximal to the original exposure, students are waiting until an externalized mechanism (i.e., the exam) prompts a need or desire for deeper understanding.

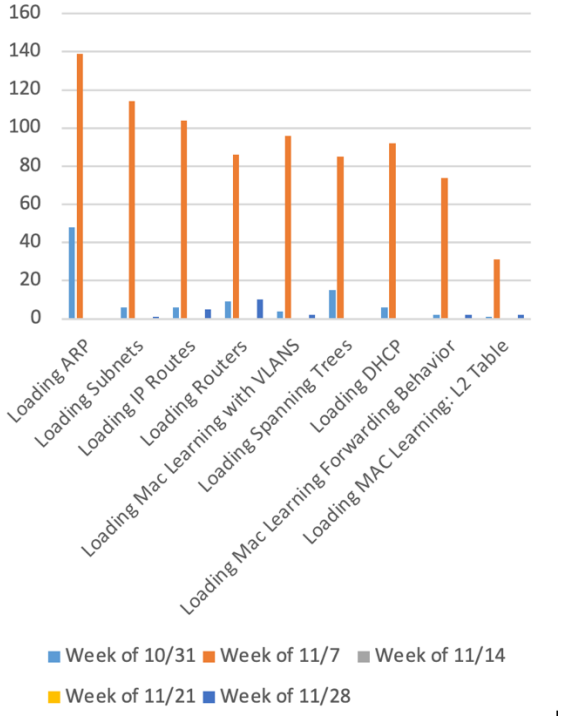


Figure 13. Practice Aggregated by Major Heading and by Week for Month of November prior to the Course Semester Exam.

In considering prior work [1, 5, 6] that underscore the importance of scaffolded learning opportunities tied closely (both in time and content) to initial exposure, this study suggests that more work may be needed to ensure that students are understanding subtopics/concepts clearly and in a way that strengthens their overarching learning possibilities. While this study serves an important purpose, it is bound by several constraints. First, it looks only at a single course in a semester that was contextualized by COVID 19. That notwithstanding, it offers interesting insight into the ways in which students engage with a connected set of complementary learning opportunities.

5.2 Implications for Practice

This study positively reinforces the utility and importance of providing multiple pathways for students to learn content material. Building on the work of this instructor, findings identify that when made available, students will engage with different forms of curricular presentation and for at least some, will revisit those opportunities multiple times. Further, the findings suggest that opportunities that are low stakes (e.g., without serious grade consequences) may be especially important in allowing students the active space needed to master concepts.

5.3 Implications for Future Research

This study lays important foundation for future research in this area. Specifically, subsequent studies might usefully understand with finer grain individualized patterns of student use connected across course learning opportunities as well as the ways in which those usage patterns connect with various outcomes (e.g., grades, satisfaction, sense of agency).

6 CONCLUSION

Work continues to be needed to ensure that we are providing and understanding the utility of various learning opportunities toward the larger academic outcomes of interest. In the field of network security, the generation of highly skilled graduates able to engage the work effectively has never been more needed. This study reminds us that the pathway to a strong workforce begins with the classroom.

ACKNOWLEDGMENTS

This material is based upon work partially supported by the National Science Foundation under Award Number SATC Education, DGE-1907537. We are grateful for their generosity; all conclusions are our own.

REFERENCES

[1] Brian R. Belland. 2017. *Instructional Scaffolding in STEM Education*. Springer Open Access. Retrieved from https://link.springer.com/book/10.1007/978-3-319-02565-0

[2] Harald Bergsteiner, Gayle Avery, and Ruth Neumann. 2010. Kolb's experiential learning model: Critique from a modeling perspective. *Studies in Continuing Education 321*, 29-46.

[3] Gerald Burch, Robert Giambatista, John Batchelor, Jana Burch, Duane Hoover, and Nathan Heller. 2019. A meta-analysis of the relationship between experiential learning and learning outcomes. *Decision Sciences 17*, 3, 239-273.

[4] Wm. Arthur Conklin, Raymond Cline, and Tiffany Roosa. 2014. Re-engineering cybersecurity education in the US: an analysis of the critical factors. In *2014 47th Hawaii International Conference on System Sciences*, January 6-9, 2014, Waikoloa, HI, 2006-2014.

[5] Ming Young Doo, Curtis Bonk, and Heeok Heo. 2020. A meta-analysis of scaffolding effects in online learning in higher education. *International Review of Research in Open and Distributed Learning 21*, 3, 60-80.

[6] Richard A. Duschl. 2019. Learning progressions: Framing and designing coherent sequences for STEM education. *Disciplinary and Interdisciplinary Science Education Research 1*, 4. https://doi.org/10.1186/s43031-019-0005-x

[7] Howard Gardner 2011. *Frames of Mind: The Theory of Multiple Intelligences.* Basic Books, New York.

[8] Jerry Gosen and John Washbush, 2004. A review of scholarship on assessing experiential learning effectiveness. *Simulation and Gaming 35*, 2, 270-393.

[9] D. Gurkan, N. Bastin, and J. Chauvot. 2022 Computer networking security education with hands on protocol trust point observations. Manuscript submitted for publication.

[10] Gary Kessler and James Ramsay. 2013. Paradigms for cybersecurity education in a homeland security program. *Journal of Homeland Security Education 2*, 35-44.

[11] Louise Lawson, Caroline Lind, Jennifer Gibson, and Kerstin Honer zu Bentrup. 2020. Do voluntary lab-based active learning sessions impact medical student knowledge? *Medical Science Educator 30*, 2, 823-831.

[12] Mark W. McElroy. 2000. Integrating complexity theory, knowledge management, and organizational learning. *Journal of Knowledge Management 4,* 3, 195-203.

[13] Andrew McGettrick, Lillian N. Cassel, Melissa Jane Dark, Elizabeth Hawthorne, and John Impagliazzo. 2014. Toward curricular guidelines for cybersecurity. In *Proceedings of the 45th ACM technical symposium on Computer science education*, March 5-8, 2014, Atlanta, GA, 81-82.

[14] Jennifer A. Moon 2004. *A Handbook of Reflective and Experiential Learning: Theory and Practice*. New York: Routledge Falmer, New York.

[15] Rodney Petersen, Danielle Santos, Matthew C. Smith, Karen A. Wetzel, and Greg Witte. 2020. *Workforce Framework for Cybersecurity (NICE Framework)*. National Institute of Standards and Technology, Gaithersburg, MD. NIST Special Publication SP 800-181r1. https://doi.org/10.6028/NIST.SP.800-181r1

[16] Fred B. Schneider. 2013. Cybersecurity education in universities (Editorial). *IEEE Security and Privacy 11*, 4, 3-4.

[17] Karl Smith, Sherri Sheppard, David Johnson, Roger Johnson. 2005. Pedagogies of engagement: Classroom-based practices. *Journal of Engineering Education 94*, 1, 87-101.

Python-Based Tools for Modeling Transport in Porous Media Columns

Boyang Lu
Illinois Institute of Technology
Chicago, Illinois
blu6@hawk.iit.edu

David Lampert
Illinois Institute of Technology
Chicago, Illinois
Dlampert1@iit.edu

ABSTRACT

The fate and transport of dissolved constituents in porous media has important applications in the earth and environmental sciences and many engineering disciplines. Mathematical models are commonly applied to simulate the movement of substances in porous media using the advection-dispersion equation. Whereas computer programs based on numerical solutions are commonly employed to solve the governing equations for these problems, analytical solutions also exist for some important one-dimensional cases. These solutions are often still quite complex to apply in practice, and therefore computational tools are still needed to apply them to determine the concentrations of dissolved substances as a function of space and time. The Python Programming Language provides a variety of tools that enable implementation of analytical solutions into useful tools and facilitate their application to experimental data. Python provides an important but underutilized tool in environmental modeling courses. This article highlights the development of a series of Python-based computing tools that can be used to numerically compute the values of an analytical solution to the one-dimensional advection-dispersion equation. These tools are targeted to graduate and advanced undergraduate courses that teach environmental modeling and the application of Python for computing.

KEYWORDS

Python, Advection-Dispersion Model, Analytical Solution, Column Experiment, Columntracer, Dispersion Coefficient, Breakthrough Curve, Jupyter Notebook, Binder, Educational Computing Tools

1 INTRODUCTION

The fate and transport of dissolved constituents in porous media has many important applications in geology, environmental science, and engineering. Field and laboratory studies are often used to study the fate and transport of contaminants in porous media. These studies also require computational tools for interpretation of data and forecasting of pollutant migration into uncontaminated areas. Since many contaminants released to the environment are eventually trapped in soils and sediments, these media can contribute to the contamination to surface water and groundwater in the vicinity, depending on the contaminant characteristics and site geological properties.

Laboratory columns are widely used to study fate and transport in porous media such as soils and sediments. For example, McKenzie et al. [13] and Høisæter et al. [8] recently conducted column experiments to improve the understanding of per- and polyfluoroalkyl substances, an emerging class of pollutants, in unsaturated soils and groundwater. Perujo et al. [16] carried out a laboratory-scale column experiment to study the interaction between physical heterogeneity and microbial processes in subsurface sediments, and Westerhoff et al. [22] performed column tests for arsenate removal in iron oxide packed bed columns. The main purpose of column experiments is to investigate the transport and attenuation of a specific compound within a specific sediment or substrate [2]. Column experiments are flexible and simple to manage; therefore, it is possible to run a column experiment as part of an educational course. The boundary conditions, physical and chemical properties of the contaminants, media characteristics, and the type of the solvent can be controlled easily during preparation. The resulting data can provide a useful educational experience for students that are learning about fate and transport modeling.

The movement of dissolved constituents in porous media strongly depends on the fluid flow characteristics. In laboratory columns, it is reasonable to assume the flow is one-dimensional. Tracer studies using an inert substance that does not interact with the media are frequently used to assess fluid flow. The results of a tracer study provide data that can be used with an appropriate model to interpret the fluid movement, which can then be used to assess migration of other substances within the media.

Mathematical models based on advection (the movement of a dissolved substance with the bulk media) and dispersion (the dissipation of concentration gradients in the media due to differences in flow path lengths) are often used to simulate the fate and transport of dissolved substances. One-dimensional advection-dispersion models often provide excellent performance in explaining observed concentrations within laboratory columns used to study the movement of dissolved substances within porous media [1]. Students in the earth and environmental sciences and engineering disciplines require substantial training in computational science to apply these models. In addition to knowledge of the underlying physical and chemical processes, these students often also require training in the solution of differential equations and the development of computer programs to perform the calculations. The Python Programming Language provides a convenient platform for solving advection-dispersion problems, since it provides access to many applicable computational and visualization tools; however, limited educational tools are available to teach the applications of Python for environmental modeling.

The one-dimensional dispersion-advection model can be used to simulate the behavior of tracer transport in porous media. An analytical solution for the model has been developed in the Fortran programming language that is described in a report published by the U.S. Geological Survey (USGS), which includes three additional useful analytical solutions to 1-dimensional dispersion-advection equation in porous media, and more solutions to 2 and 3-dimensional situations [23]. Fortran is still used today for high performance computing, but it is difficult to implement for analytical

DOI: https://doi.org/10.22369/issn.2153-4136/14/1/2

solutions. High-level languages like Python provide many external libraries for specific needs such as root finding, minimization, and graphics that make it a more suitable alternative on modern computing platforms for problems such as the transport of a tracer in a porous medium.

In this article, a new Python library "columntracer" and a suite of supplementary Jupyter Notebooks [11] illustrating its development and usage are presented. The software is written entirely in Python and is freely available online. Key features of columntracer include the ability to: (1) calculate the solute concentration at any point in time and space in a column, (2) plot concentration profiles and breakthrough curves, and (3) fit experimental data at the outlet to breakthrough curves to find dispersion coefficient. In experimental column studies, effluent concentrations are easily obtained, while the dispersion coefficients are a key unknown parameter. By using columntracer, dispersion coefficients can be determined with a few lines of code. This library and the associated Jupyter Notebooks serve as a potentially useful educational tool for students in environmental modeling classes. Students are able to learn how contaminants flow through the column, how different initial conditions affect the concentration profile across the column and lead to different final concentrations at outlet, and how to fit experimental data to obtain dispersion coefficient of the process. The Notebooks also demonstrate the potential power of using Python versus computing tools that are more familiar to environmental science and engineering students, such as spreadsheet programs.

This project was conducted as an individual special problem for three credit hours for the student lead author. The objectives of the project were to: (1) deepen student understanding of the modeling of fate and transport of contaminants in porous media, (2) improve student Python programming skills, including creating Python classes, utilizing modules, managing code on GitHub, and publishing the columntracer library, and (3) provide an alternative to the U.S. Geological Survey Fortran based program for solving the dispersion-advection equation by developing a Python implementation.

2 METHOD

2.1 Model Description

Consider a cylindrical column of length L with flow entering on one end and exiting on the other end. The velocity of the flow U is easily measured by monitoring the flow rate (volume that exits per unit time). The dispersion coefficient D represents the tendency of the concentration gradients to dissipate. Tracer experiments using conservative substances such as bromide are typically used, along with a model, to estimate this parameter. The solute concentration in the influent for a tracer has a constant concentration of C0, and eventually the concentration leaving the column will also have a concentration of C0, at which point the tracer is said to have achieved "breakthrough." Before breakthrough, the concentration in the effluent gradually increases from zero and to the influent concentration level. Figure 1 illustrates the model system for the case when $C_0 = 100$, $L = 30$, $U = 10$, and $D = 100$. The parameters that affect the output from the column for this model are listed in .

.

Table 1. List of model parameters.

Parameters	Description	Units
C_0	Solute influent concentration	mg/L
U	Flow velocity in column	cm/hr
D	Dispersion coefficient	cm^2/hr
L	Length of column	cm

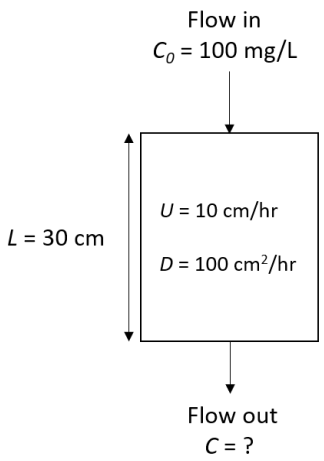


Figure 1. Schematic of Column Mode.

2.2 Advection and Dispersion Equation

2.2.1 Model and Boundary Conditions

Equation (2.1) shows the model used in the software, which is the one-dimensional advection-dispersion equation. The value of C is the concentration at time t and distance x from the inlet. The equation is based on a material balance within a differential element, and it assumes constant value of the parameters U and D.

$$\frac{dC}{dt} = D\frac{d^2C}{dx^2} - U\frac{dC}{dx} \tag{2.1}$$

Two boundary conditions and one initial condition are required to solve the equation. Assuming there is no tracer in the column at the start of the simulation, the initial condition is zero concentration, as shown in Equation (2.2). The boundary condition for the influent is flux-matching (i.e., the mass flow of the tracer into the column equals the mass flow inside the column at $x = 0$). The advective flux into the column matches the advective and dispersive fluxes at the start of the column in Equation (2.3). The Danckwerts' boundary condition used at the effluent assumes that the dispersion flux is negligible, meaning the derivative is zero, which is shown in Equation (2.4) [25].

$$C(x, t = 0) = 0 \tag{2.2}$$

$$UC_0 = UC(x = 0, t) - D\frac{dC(x = 0, t)}{dx} \tag{2.3}$$

$$\frac{dC(x = L, t)}{dx} = 0 \tag{2.4}$$

The equations can be non-dimensionalized using the dimensionless time t^*, distance x^*, and concentration C^*, which simplifies the mathematics as shown in Equations (2.5)-(2.7). In the dimensionless domain, the three parameters are reduced to just one parameter, defined as the Peclet number Pe in Equation (2.8), which represents the ratio of the importance of advection to dispersion processes in the column.

$$t^* = \frac{Dt}{L^2} \tag{2.5}$$

$$x^* = \frac{x}{L} \tag{2.6}$$

$$C^* = \frac{C}{C_0} \tag{2.7}$$

$$Pe = \frac{UL}{D} \tag{2.8}$$

The governing equation, initial condition, and boundary conditions become Equations (2.9)-(2.12), after normalization.

$$\frac{dC^*}{dt^*} = \frac{d^2C^*}{dx^{*2}} - Pe\frac{dC^*}{dx^*} \tag{2.9}$$

$$C^*(x^*,\ t^* = 0) = 0 \tag{2.10}$$

$$C^*(x^* = 0,\ t^*) - \frac{1}{Pe}\frac{dC^*(x^* = 0,\ t^*)}{dx^*} = 1 \tag{2.11}$$

$$\frac{dC^*(x^* = 1, t^*)}{dx^*} = 0 \tag{2.12}$$

2.2.2 Analytical Solution to the Model

The dimensionless governing equation and auxiliary conditions (Equations (2.9)-(2.12)) are a boundary value problem that can be solved using separation of variables [18]. The USGS summarizes four analytical solutions for the 1-dimensional advection-dispersion equation, including the model problem shown in Equation (2.13, which is referred to as a "Finite System with Third-Type Source Boundary Condition" in the report. Only one analytical solution is included in columntracer, but the others could be added easily in the future or developed for other student projects. Furthermore, analytical solutions to 2 and 3-dimensional problems in different situations are also available [23]. The values of β_i are the eigenvalues of the boundary value problem, and the corresponding terms in the infinite series are the eigenfunctions [5]. The eigenvalues are determined by finding the roots of Equation (2.14), which is the characteristic equation of the boundary value problem.

$$C^*(x^*,\ t^*) = 1 - 2Pe \cdot e^{\left(\frac{Pe}{2}x^* - \frac{Pe^2}{4}t^*\right)} \cdot \sum_{n=1}^{\infty} \frac{\beta_i\left[\beta_i\cos(\beta_i x^*) + \frac{Pe}{2}\sin(\beta_i x^*)\right]}{\left[{\beta_i}^2 + \frac{Pe^2}{4} + Pe\right]\left[{\beta_i}^2 + \frac{Pe^2}{4}\right]} \cdot e^{-\beta_i^2 t^*} \tag{2.13}$$

$$\beta cot\beta - \frac{\beta^2}{Pe} + \frac{Pe}{4} = 0 \tag{2.14}$$

A sufficient number of eigenvalues must be estimated to perform the summation in Equation (2.13). The characteristic equation (2.14) has no exact solution, unlike some other characteristic equations commonly encountered in diffusion boundary value problems. The eigenvalues for a given column system with parameters U, D, and C_0 depend only on the Peclet number defined in Equation (2.8). The values for a given Pe can be determined by finding the roots of Equation (2.15). The function has a singularity at all integral multiples of π based on trigonometric relationships shown in Equation (2.16) and (2.17).

$$F(Pe,\beta) = \beta cot\beta - \frac{\beta^2}{Pe} + \frac{Pe}{4} \tag{2.15}$$

$$cot\beta = \frac{cos\beta}{sin\beta} \tag{2.16}$$

$$sin\beta = 0\ at\ \beta = 0, \pi, 2\pi, \ldots = n\pi \tag{2.17}$$

Between each singularity, the function has exactly one zero. Figure 2 shows the value of the function across the first ten singularities. It also shows the first few roots. In Figure 2, the value of the function, the singularities at every $n\pi$, and the location of the first eigenvalue near $\beta = 1.54$ can be seen. To use the model result, the first task is to identify the eigenvalue across each interval. Scientific Python (SciPy) has an optimization library with a variety of methods to determine the root of a function. For the model problem, Brent's method [3] can be used to solve the characteristic equation.

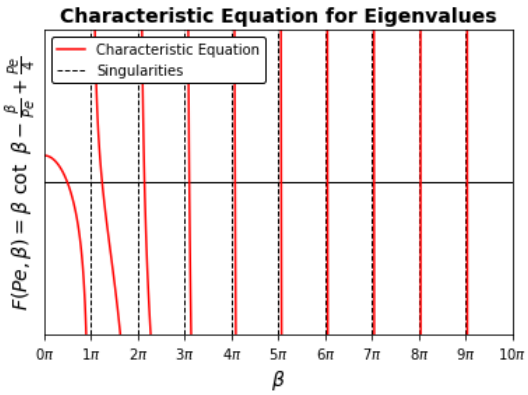


Figure 2. Characteristic Equation for Eigenvalues.

Brent's method (also sometimes called the van Wijngaarden-Dekker-Brent method) is a root-finding algorithm which combines root bracketing, bisection, and inverse quadratic interpolation. It uses a Lagrange interpolating polynomial of degree 2. Brent [3] claims that this method always converges as long as the values of the function are computable within a given region containing a root. Given three points x_1, x_2, and x_3, Brent's method fits x as a quadratic function of y, then uses the interpolation formula described in Equation (2.18) [21].

$$x = \frac{[y - f(x_1)][y - f(x_2)]x_3}{[f(x_3) - f(x_1)][f(x_3) - f(x_2)]} + \frac{[y - f(x_2)][y - f(x_3)]x_1}{[f(x_1) - f(x_2)][f(x_1) - f(x_3)]} + \frac{[y - f(x_3)][y - f(x_1)]x_2}{[f(x_2) - f(x_3)][f(x_2) - f(x_1)]} \tag{2.18}$$

Subsequent root estimation is obtained by setting y = 0, giving

$$x = x_2 + \frac{P}{Q} \tag{2.19}$$

where P and Q are:

$$P = S[T(R - T)(x_3 - x_2) - (1 - R)(x_2 - x_1)] \tag{2.20}$$

$$Q = (T - 1)(R - 1)(S - 1) \tag{2.21}$$

with

$$R \equiv \frac{f(x_2)}{f(x_3)} \tag{2.22}$$

$$S \equiv \frac{f(x_2)}{f(x_1)} \tag{2.23}$$

$$T \equiv \frac{f(x_1)}{f(x_3)} \tag{2.24}$$

Following the determination of a suitable number of eigenvalues, the simulated concentration is computed at any point in the domain by summing the eigenvalues in Equation (2.13).

2.2.3 Dispersion Coefficient Fitting

One of the primary applications of Equation (2.13) is to determine the value of the dispersion coefficient in the media. The velocity and initial concentration can be measured relatively easily for a

given experiment. Determining the value of D requires an inverse parameter fitting, which typically requires running many simulations, assessing performance, and optimizing an objective function, such as minimizing error. A common approach for parameter estimation is to compare the model simulation with experimental results. Given a set of values for the effluent concentration at $x = L$ at various points in time, a series of values of the dispersion coefficient can be used to calculate the concentrations corresponding to specific data points of breakthrough curve.

After the simulated concentrations corresponding to each data point are calculated, the mean squared error (MSE) between the simulated data and the experimental data can be determined. The MSE is calculated using Equation (2.25), where $\widehat{C}_i$ is the simulated concentration, C_i is the measured concentration, n is the number of measurements. The goal of fitting process is to minimize the MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\widehat{C}_i - C_i)^2 \tag{2.25}$$

Scientific Python (SciPy) offers several functions for minimization in its "optimize" module. Four different functions are available, depending on the nature of the application. The default function is fmin, which uses the downhill simplex algorithm, also known as the Nelder-Mead method [14]. The other options are fmin_powell, fmin_cg, and fmin_bfgs, which use Powell's method [17], the nonlinear conjugate gradient algorithm [15], and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [6], resp. The results of different functions were compared for performance as described in Section 3.3. One of the advantages of using Python is the ready availability of these tools for applications such as parameter fitting. Demonstrating these capabilities to students can help to close the gap and develop computational thinking skills for students with limited programming experience.

After the dispersion coefficient is determined, the coefficient of determination, R^2, can be calculated using Equation (2.26), where $\bar{C}_s$ and $\bar{C}_m$ represent the mean values of the model, $\widehat{C}_i$, and the observations, C_i, resp.

$$R^2 = \frac{(\sum_{i=1}^{i=n}(\widehat{C}_i - \bar{C}_s)\cdot(C_i - \bar{C}_m))^2}{\sum_{i=1}^{i=n}(\widehat{C}_i - \bar{C}_s)^2 \cdot \sum_{1}^{n}(C_i - \bar{C}_m)^2} \tag{2.26}$$

2.3 Software

2.3.1 Description

The model tools described in Section 2.2 have been compiled into a software library known as "columntracer." The columntracer library is written completely in Python [19]. Python is an interpreted high-level open-source programming language, with a design philosophy that emphasizes code readability. Python's user-friendly syntax and interpreted nature decrease the time requirements for new users (e.g., students in an environmental modeling course) to begin applying the software to problem solving. Python's extensibility and interpreted nature allow new users to perform complex tasks by integrating various libraries, thereby saving time [12]. The key third-party modules used in columntracer are Numeric Python (NumPy) for generating and calculating arrays and matrixes [7], Scientific Python (SciPy) for optimizing and solving equations [20], and the Math Plotting Library (Matplotlib) for plotting and visualization [9].

The Jupyter Notebook is an open-source web application that allows user to create and share documents that contain live code, equations, visualizations, and narrative texts [11]. Jupyter Notebooks have been generated to illustrate the computation procedure outlined in Section 2, show applications of the columntracer classes that enable rapid development of new models, and provide documentation of the source code that is available on GitHub [4]. In the documentation, examples are demonstrated with both code and narrative texts. An example data set for fitting the dispersion coefficient is also available in the repository that comes with the module. The data set was taken from a study that compared the performance of different models that were fit to experimental concentrations in a one-dimensional column [24].

The Binder project offers an easy place to share computing environment to everyone. It allows project creators to specify custom environments and share them with a single link [10]. With the link, users are able to get access to the project without downloading any required software or packages. On the columntracer GitHub page [4], Binder links are provided for interacting the Jupyter Notebooks, requiring only a web browser. With the help of Binder, it's easy to demonstrate the functions of columntracer in classroom or other educational situations that Python environment is not immediately accessible.

2.3.2 Walk-through

Figure 3 shows a screenshot of the code for importing the columntracer package and performing a demo run with the default parameters.

```
from columntracer import ColumnTracer
```

```
%matplotlib inline

c = ColumnTracer(demo = True,
                 demo_plot = True,
                 demo_plot_save = False)
```

```
Default parameters for the demo are:
solute influent concentration C0 = 100 mg/L,
flow velocity in column U = 10 cm/h,
dispersion coeffiecient D = 100 cm2/h,
length of column L = 30 cm,
number of terms to use in series solution n = 1000.
```

Figure 3. Example Code for Import and Demo Run

Four methods are called during the demo run in the following sequence: (1) "characteristic_equation" that computes and plots the characteristic equation for a given *Pe* (3 in the demo), (2) "eigenvalues" that calculates the first *n* eigenvalues (1000 in the demo), (3) "concentration_profile" that calculates the concentrations across the column at various times (0.00001, 0.1, 0.5, 1, 2, 4, and 10 hours in the demo), and (4) "effluent_concentration" that calculates the concentration at the outlet of the column (0 to 12 hours in the demo).

By setting the parameter "demo_plot" to True, the software generates plots of the characteristic equation for eigenvalues (Figure 2), the column concentration profiles (Figure 11), and the column breakthrough curve (Figure 12), which can also be obtained by the code in Figure 4, Figure 6, and, Figure 8, respectively. The parameter "demo_plot_save" determines whether to save the plots to a local file, and "savefig_dpi" specifies the image quality (200 dots per inch, dpi).

The parameter "savefig" in Figure 4, Figure 6, and, Figure 8 controls the export of plots, and it is set to False by default. Setting savefig to True will save the plot to the working directory with a default file names of "characteristic_equation," "concentration_profile," and "breakthrough_curve," respectively. Users can also assign a string to the parameter to name the image files. If users

want an image with lower or higher quality, they can change the value of parameter "savefig_dpi."

```
%matplotlib inline

c = ColumnTracer(C0 = 100,
                 U = 10,
                 D = 100,
                 L = 30,
                 n = 1000)

c.characteristic_equation(plot = True,
                          savefig = False,
                          savefig_dpi = 200)
```

Figure 4. Example Code to Compute the Characteristic Equation.

```
c = ColumnTracer(C0 = 100,
                 U = 10,
                 D = 100,
                 L = 30,
                 n = 1000)

eff = c.get_concentration(time = 9,
                          x = 1)
print('The concentration is {:.2f} mg/L'.format(eff))
```

```
The concentration is 98.23 mg/L
```

Figure 5. Example Code for Calculating the Concentration at a Given Time and Position.

Columntracer can calculate the concentration at any given time and location in the column as shown in Figure 5. The *x* values range from 0 to 1, which is similar to the parameter "position" in Figure 6, indicating the location from the beginning to the end of the column. In Figure 5, the effluent concentration at 9 hours is calculated to be 98.23 mg/L. By calculating the concentration throughout the column at a given time, a concentration profile can be created as shown in Figure 6.

```
%matplotlib inline

c = ColumnTracer(C0 = 100,
                 U = 10,
                 D = 100,
                 L = 30,
                 n = 1000)

t = [0.00001, 0.1, 0.5, 1, 2, 4, 10]
pos = [0, 0.01, 0.02, 0.04, 0.06,
       0.08, 0.1, 0.15, 0.2, 0.25,
       0.3, 0.4, 0.5, 0.6, 0.7, 0.8,
       0.85, 0.9, 0.95, 0.98, 0.99, 1]
c_profile = c.concentration_profile(times = t,
                                    positions = pos,
                                    plot = True,
                                    print_conc = False,
                                    savefig = False,
                                    savefig_dpi = 200)
```

Figure 6. Example Code for Calculating and Plotting Concentration Profiles.

In Figure 6, concentration profiles are calculated for t = 0.00001, 0.1, 0.5, 1, 2, 4, and 10 hours at different locations through the column, which are indicated by the variable "pos." The parameter "positions" must be provided as a list of values ranging from 0 to 1, and each value represents the ratio of the distance in the column to the total length of the column. This method returns a list of concentration lists that can be printed by setting parameter "print_conc" to True. Each concentration list corresponds to a time in the parameter "times," and each list has the same length as parameter "positions." By using the concentration stored in variable "c_profile," users can access the data and make plots using the Matplotlib package [9] as illustrated in Figure 7.

For calculating and plotting a breakthrough curve such as the one shown in Figure 8, users must provide a time period for the solute transport, as well as the time interval, which determines how many data points are calculated. The parameter "time_start" is 0 by default, but can modified if a different starting time is desired. This method returns a list of concentrations that can be used for printing or plotting. Users can also choose to use automatic plotting by setting parameter "plot" to True, or create plots manually as shown in Figure 9.

```
import matplotlib.pyplot as plt
# plotting
fig, ax = plt.subplots()
ax.set_xlabel('Position in column (cm)',
              size = 12, weight = 'bold')
ax.set_ylabel('Concentration (mg/L)',
              size = 12, weight = 'bold')
ax.set_title('Column Concentration Profiles',
             size = 14, weight = 'bold')

for t, cs in zip(default_t, c_profile):
    ax.plot(default_pos, cs, label = 't = {:.1f}h'.format(t))
ax.legend(loc = 'right', bbox_to_anchor = (1.4, 0.5),
          fontsize = 12)
```

Figure 7. Example Code for Accessing Concentration Profile Data after Numerical Computation.

For calculating and plotting a breakthrough curve such as the one shown in Figure 8, users must provide a time period for the solute transport, as well as the time interval, which determines how many data points are calculated. The parameter "time_start" is 0 by default, but can modified if a different starting time is desired. This method returns a list of concentrations that can be used for printing or plotting. Users can also choose to use automatic plotting by setting parameter "plot" to True, or create plots manually as shown in Figure 9.

```
%matplotlib inline

c = ColumnTracer(C0 = 100,
                 U = 10,
                 D = 100,
                 L = 30,
                 n = 1000)
time_start = 0
time_end = 12
interval = 0.1
Cs = c.effluent_concentration(time_end = time_end,
                              interval = interval,
                              time_start = time_start,
                              plot = True,
                              print_conc = False,
                              savefig = False,
                              savefig_dpi = 200)
```

Figure 8. Example Code for Calculating the Effluent Concentration and Plotting the Breakthrough Curve.

In Figure 10, a csv file containing time and effluent concentration data is imported. The first 8 values in the file are also shown in the figure. The data are used to fit to a breakthrough curve, so that a dispersion coefficient can be determined. The data processing

in the figure is only for the example data set, which is a csv file with 2 columns: one for time and the other for the corresponding concentrations. The csv file is available in the columntracer module folder or can be accessed on GitHub repository [4]. The initial concentration, solute velocity, length of the column, and the initial guess of the dispersion coefficient are required for the dispersion coefficient fitting. Four algorithms are available for minimization, which are described in Section 3.3. Setting the parameter "algorithm" to None applies the default algorithm: the Nelder-Mead method.

```
import numpy as np
import matplotlib.pyplot as plt
# plotting
time = np.arange(time_start, time_end, interval)

fig, ax = plt.subplots()
ax.set_xlabel('Time (hr)',
              size = 12, weight = 'bold')
ax.set_ylabel('Concentration (mg/L)',
              size = 12, weight = 'bold')
ax.set_title('Column Breakthrough Curve',
             size = 14, weight = 'bold')
ax.plot(time, Cs, label = 'Breakthrough curve',
        ls = '-', c = 'r')
# plug flow line
xs = [0, L/U, L/U, time_end]
ys = [0, 0, C0, C0]
ax.plot(xs, ys,
        ls = '-.', lw = 1, c = 'b', label = 'Plug flow')
ax.legend()
```

Figure 9. Example Code for Accessing Effluent Concentration Data after Numerical Computation.

```
# use pandas to help read data from external csv file
import pandas as pd
import sys

# example data file is available in
# \Lib\site-packages\columntracer
# it's also available in the package repository:
# https://github.com/BYL4746/columntracer
path = sys.executable.split('python.exe')[0]
       + 'Lib\\site-packages\\columntracer\\'
data = pd.read_csv(path + 'data.csv')

# convert pandas DataFrame to lists
fit_t = data['time'].values.tolist()
fit_c = data['concentration'].values.tolist()

# fit D with known U
c = ColumnTracer(C0 = 1,
                 U = 34,
                 L = 650,
                 n = 1000)

result = c.fit_D(time = fit_t,
                 conc = fit_c,
                 algorithm = None,
                 initial_guess=175,
                 plot = True)
```

time	concentration
17.35043	0.102756522
17.5641	0.147771513
17.75641	0.239908041
18.39744	0.313172909
18.61111	0.363423502
18.84615	0.477547322
19.10256	0.625177876
19.33761	0.654484942

Figure 10. Example Code for Fitting Data to Breakthrough Curve to Fit the Dispersion Coefficient.

3 RESULTS

Examples of several model applications are provided in Jupyter Notebooks that describe the code and show plots of the output for educational purposes. A general description of the model and a detailed set of examples scripts for columntracer are provided with the source repository [4]. Jupyter Notebooks are recommended for educational applications, but other Python environments can also be used, including the Command Prompt or the IPython console. Alternative text editors and integrated development environments (IDE) such as PyCharm and Spyder can also be used to work with the code, particularly since columntracer is provided as a library on the Python Package Index (PyPI) [26].

3.1 Concentration Profiles

After the eigenvalues for a given parameter set (*Pe*) have been determined, the concentration can be evaluated at any point in time and space with the same approach described in Section 2.2.2. Figure 11 shows the evolution in the concentration profile over time throughout the column for the example case where initial concentration $C_0 = 100$ mg/L, the column length $L = 30$ cm, the solute velocity $U = 10$ cm/hr, and the dispersion coefficient $D = 100$ cm^2/hr. The number of eigenvalues used for the example was $n = 1000$. At the beginning of the simulation, the concentration is zero everywhere, as expected, while at the end, the concentration has equilibrated with the influent concentration throughout the column. Between these extremes, the concentration gradually increases throughout the column.

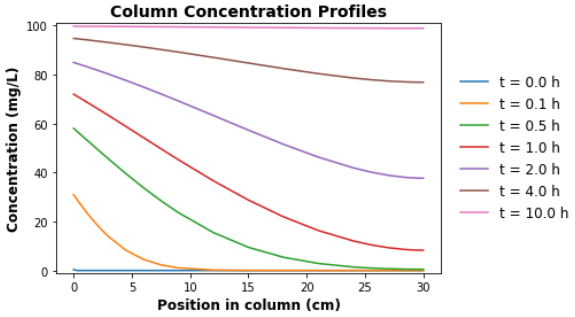


Figure 11. Column Concentration Profiles for t = 0, 0.1, 0.5, 1.0, 2.0, 4.0, 10 hours.

3.2 Breakthrough Curve

The concentration at the outlet is of primary interest for tracer studies, since it can be compared to observed data. A high-resolution time series of concentrations can easily be obtained by evaluating the function at the outlet ($x = L$), and the breakthrough curve is shown in Figure 12. The dotted line in blue indicates the breakthrough with $D = 0$, which is known as "plug flow," since the fluid flow paths in this case are all the same causing flow in a "plug" motion.

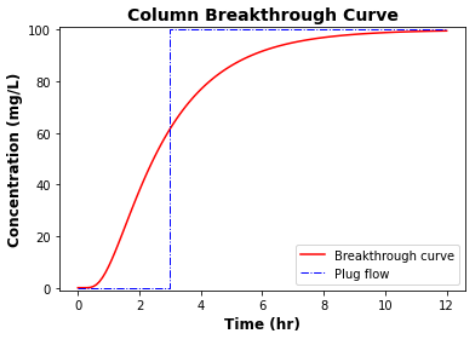


Figure 12. Column Breakthrough Curve.

3.3 Dispersion Model Fitting

Experimental data for a column were obtained from Xiong et al. [24] and are provided with the columntracer repository on GitHub

[4]. The column length in the study was 650 cm, and the velocity was 34 cm/hr. The concentrations in this publication were non-dimensionalized, meaning they represent the dimensionless C^* (the ratio of the effluent to the influent concentration), so the values range between 0 and 1. By choosing the default minimization method from SciPy, fmin, which uses Nelder-Mead method, and with an initial guess of 175 cm^2/hr, the fitted dispersion coefficient is 193.9 cm^2/hr, with mean squared error (MSE) of 0.0888 and R^2 of 0.964. Figure 13 shows the raw data and the breakthrough curve based on fitted dispersion coefficient. In Xiong et al. [25], the coefficient was fitted to be 74 cm^2/hr with root mean square error (RMSE) of 0.0313 and an R^2 of 0.9935. The relatively higher MSE may be caused by the inaccuracy and low quantity of the experimental data obtained from the figure. Because porosity η was not considered in the software, the adjusted dispersion coefficient is calculated to be 77.56 cm^2/hr using Equation (3.1), assuming $\eta = 0.4$, which is in good agreement with the value determined from the original analysis.

$$D_{adj} = \eta D \tag{3.1}$$

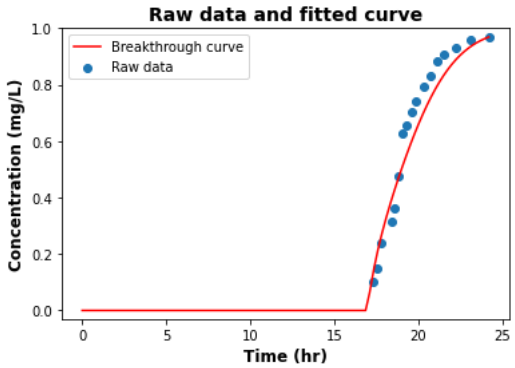


Figure 13. Fitted Breakthrough Curve and Experimental Data.

The other three minimization methods from SciPy were also tested. The results are listed in Table 2, which compare the time consumption to determine D for the different algorithms. For the sample dataset, both the Nelder-Mead method and Powell's method achieved satisfactory MSEs, whereas the MSEs for the nonlinear conjugate gradient algorithm and BFGS algorithm were less satisfactory. Students can easily try different algorithms to find the most suitable approach for their own data sets in an educational environment.

Table 2. Fitted Dispersion Coefficient and MSE Using Different Minimization Algorithms.

Function	Algorithm	*D* (cm^2/hr)	MSE (-)	Time (s)
fmin	Nelder-Mead method	193.9	0.089	4.73
fmin_powell	Powell's method	171.2	0.088	19.1
fmin_cg	Nonlinear conjugate gradient algorithm	175	0.18	7.7
fmin_bfgs	BFGS algorithm	175	0.19	9.6

3.4 Verification, Validation, and Accreditation

The USGS report provides an example (Sample Problem 2) in section titled "Finite System with Third-Type Source Boundary Condition" that includes detailed computational results shown in attachment 4 [23]. The input data from this sample problem were used for validation of the columntracer library. The input parameters include: the initial concentration $C_0 = 1$ mg/L, the column length $L = 12$ inches (30.48 cm), the solute velocity $U = 0.6$ inch/hr (1.524 cm/hr), and the dispersion coefficient $D = 0.6$ $inch^2$/hr (3.87096 cm^2/hr). The concentration profiles are shown in Figure 14, in which the lines represent the simulated data by columntracer and markers represent the data provided in the USGS report. The detailed results are presented in U.S. Customary units in APPENDIX in the Appendix and show perfect consistency with the results in the report.

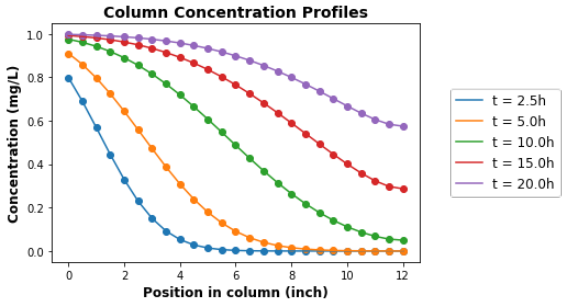


Figure 14. Concentration Profiles for Sample Problem 2.

The simulation results in Table 3 and the concentration profile plot are provided in the USGS report on pages 219 and 31, respectively. Both Figure 14 and Table 3 show a perfect reproduction of the results from the USGS program using the columntracer simulation. The time consumption is about 0.03 seconds, which is over a hundred times faster than the FORTRAN program described in the report. A Jupyter Notebook is available showing these results in a folder called "validation" on GitHub repository for validation [4]. The results provide a comparison of the utility of the modern interactive Python notebook and a compiled and less user-friendly FORTRAN program from 30 years ago.

The source code, data, and Jupyter Notebooks can be found on GitHub repository [4]. The columntracer library is downloadable from PyPI by using command "pip install columntracer" in Command Prompt. With the installation of columntracer and a clone of the repository, users are able to verify the algorithms of the program and use the Jupyter Notebooks to validate the results of the model that are provided in the repository. The source code was installed on new machines and used to validate the results shown in this article. The results are shown with both MSE and R2 to ensure that the model correctly predict the transport process. With the help of the Binder link provided in the source repository, the Jupyter Notebooks can be run through a server online without installing any software locally. The examples provided throughout this exercise, and additional examples that were run using different parameter values all validate the model. The expected behavior that is observed in physical experiments is reproduced from the model results.

The information in this article is useful for both undergraduate and graduate students in environment-related majors, teachers who teach courses involving fluid transport in porous media, and researchers who perform column experiments. The information provided can help students both be trained with Python programming ability and learn modeling of fate and transport of dissolved constituents.

4 CONCLUSION

Column experiments are useful for studying fate and transport of solutes through porous media. A new open-source software tool,

columntracer, has been developed to help user better understand the column experiment. The software provides solutions to advection-dispersion equation as well as the visualization of the solutions, which includes plotting the characteristic equation, concentration profiles, and the effluent breakthrough curve. The software can also be used to fit the dispersion coefficient using experimental effluent data by minimizing the mean-squared error. The columntracer library can be a useful tool for research, but it is also appropriate for educational purposes. Students in environmental modeling courses could use the software to learn about solute transport, Python scripting, NumPy, SciPy, Matplotlib, and Jupyter Notebook by using the software and the supplementary Notebooks. The code and Notebooks are open-source and freely available online [4].

5 REFLECTION

By completing the project, I managed to learn how to define and modify a Python class by manipulating attributes and functions, as well as by implementing third-party libraries including NumPy, SciPy, Matplotlib, Jupyter, and Binder. In addition to programming skills, I became acquainted with the advection-dispersion model, and its analytical solution solved by separation of variables. I also learned how to perform parameter fitting using optimization within the Python environment. During the acquirement of these skills, there were several challenges that I faced. Debugging was one of the hardest, because sometimes a typo could result in a break-down or an unexpected outcome. Finding an appropriate method for the parameter fitting was also challenging, because there were numerous approaches available. On the whole, the project improved both my programming ability and specialized knowledge in my major, and would help me in future projects such as programming for wastewater process optimization and machine learning in Python. For these reasons, I consider the project an overall success.

REFERENCES

[1] Munshoor Ahmed, Qurat Ul Ain Zainab, and Shamsul Qamar. 2017. Analysis of One-Dimensional Advection–Diffusion Model with Variable Coefficients Describing Solute Transport in a Porous medium. *Transp Porous Med 118*, 3 (July 2017), 327–344. DOI:https://doi.org/10.1007/s11242-017-0833-0

[2] Stefan Banzhaf and Klaus Hebig. 2016. Use of column experiments to investigate the fate of organic micropollutants - A review. *Hydrology and Earth System Sciences 20*, (September 2016), 3719–3737. DOI:https://doi.org/10.5194/hess-20-3719-2016

[3] Richard P. Brent. 2013. *Algorithms for Minimization Without Derivatives*. Courier Corporation.

[4] BYL4746. 2021. BYL4746/columntracer. Retrieved July 7, 2021 from https://github.com/BYL4746/columntracer

[5] R. Courant and D. Hilbert. 1989. *Methods of Mathematical Physics* (1st ed.). Wiley, New York, New York. DOI:https://doi.org/10.1002/9783527617210

[6] Roger Fletcher. 1987. *Practical Methods of Optimization*. Wiley, New York, New York. Retrieved June 29, 2021 from http://archive.org/details/ practicalmethods0000flet

[7] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature 585*, 7825 (September 2020), 357–362. DOI:https://doi.org/10.1038/s41586-020-2649-2

[8] Åse Høisæter, Anja Pfaff, and Gijs D. Breedveld. 2019. Leaching and transport of PFAS from aqueous film-forming foam (AFFF) in the unsaturated soil at a firefighting training facility under cold climatic conditions. *Journal of Contaminant Hydrology 222*, (April 2019), 112–122. DOI: https://doi.org/10.1016/j.jconhyd.2019.02.010

[9] John D. Hunter. 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering 9*, 3 (May 2007), 90–95. DOI: https://doi.org/10.1109/MCSE.2007.55

[10] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, M. Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan-Kelley, and Carol Willing. 2018. Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. In *Proceedings of the 17th Python in Science Conference (2018)*, July 9-15, 2018, Austin, Texas, 113–120. DOI:https://doi.org/10.25080/Majora-4af1f417-011

[11] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Prez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damian Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. 2016. Jupyter Notebooks – a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016), 87–90. DOI:https://doi.org/10.3233/978-1-61499-649-1-87

[12] David Lampert. 2020. An introduction to Python programming for environmental professionals. *EM: Air and Waste Management Association's Magazine for Environmental Managers 2020*, (February 2020), 11–14.

[13] Erica R. McKenzie, Robert L. Siegrist, John E. McCray, and Christopher P. Higgins. 2015. Effects of chemical oxidants on perfluoroalkyl acid transport in one-dimensional porous media columns. *Environ. Sci. Technol. 49*, 3 (February 2015), 1681–1689. DOI:https://doi.org/10.1021/es503676p

[14] J. A. Nelder and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal 7*, 4 (January 1965), 308–313. DOI:https://doi.org/10.1093/comjnl/7.4.308

[15] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (2nd ed ed.). Springer, New York.

[16] N. Perujo, X. Sanchez-Vila, L. Proia, and A.M. Romaní. 2017. Interaction between physical heterogeneity and microbial processes in subsurface sediments: A laboratory-scale column experiment. *Environ. Sci. Technol. 51*, 11 (June 2017), 6110–6119. DOI:https://doi.org/10.1021/acs.est.6b06506

[17] M. J. D. Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal 7,* 2 (January 1964), 155–162. DOI:https://doi.org/10.1093/comjnl/7.2.155

[18] Bernd S. W. Schröder. 2007. *Mathematical Analysis: A Concise Introduction.* John Wiley & Sons, Inc., Hoboken, NJ. DOI:https://doi.org/10.1002/9780470226773

[19] Guido Van Rossum. 2007. Python programming language. The Guru is in session, at *USENIX Annual Technical Conference*, Santa Clara, CA.

[20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods 17,* 3 (March 2020), 261–272. DOI:https://doi.org/10.1038/s41592-019-0686-2

[21] Eric W. Weisstein. Brent's Method. From MathWorld--A Wolfram Web Resource. Retrieved June 14, 2021 from https://mathworld.wolfram.com/BrentsMethod.html

[22] Paul Westerhoff, David Highfield, Mohammad Badruzzaman, and Yeomin Yoon. 2005. Rapid small-scale column tests for arsenate removal in iron oxide packed bed columns. *J. Environ. Eng. 131*, 2 (February 2005), 262–271. DOI:https://doi.org/10.1061/(ASCE)0733-9372(2005)131:2(262)

[23] Eliezer J. Wexler. 1992. *Analytical Solutions for One-, Two-, and Three-dimensional Solute Transport in Ground-water Systems with Uniform Flow*. U.S. Government Printing Office. Retrieved from https://pubs.usgs.gov/of/1989/0056/report.pdf

[24] Yunwu Xiong, Guanhua Huang, and Quanzhong Huang. 2006. Modeling solute transport in one-dimensional homogeneous and heterogeneous soil columns with continuous time random walk. *Journal of Contaminant Hydrology 86*, 3 (August 2006), 163–175. DOI:https://doi.org/10.1016/j.jconhyd.2006.03.001

[25] 1995. Continuous flow systems. Distribution of residence times. *Chemical Engineering Science 50,* 24 (December 1995), 3857–3866. DOI:https://doi.org/10.1016/0009-2509(96)81811-2

[26] PyPI · The Python Package Index. PyPI. Retrieved July 7, 2021 from https://pypi.org/

APPENDIX

Table 3. Solute Concentration as a Function of Time for Sample Problem 2

Position in Column (inch)	Time (hr)				
	2.5	5.0	10.0	15.0	20.0
	Solute Concentration (mg/L)				
0.0	0.79858	0.90992	0.97530	0.99197	0.99716
0.5	0.68921	0.85904	0.96098	0.98727	0.99549
1.0	0.56799	0.79673	0.94230	0.98097	0.99322
1.5	0.44466	0.72419	0.91871	0.97276	0.99021
2.0	0.32919	0.64364	0.88977	0.96231	0.98629
2.5	0.22958	0.55821	0.85524	0.94926	0.98128
3.0	0.15033	0.47151	0.81509	0.93331	0.97499
3.5	0.09217	0.38726	0.76955	0.91415	0.96720
4.0	0.05280	0.30880	0.71911	0.89156	0.95771
4.5	0.02820	0.23875	0.66455	0.86537	0.94630
5.0	0.01402	0.17878	0.60686	0.83551	0.93276
5.5	0.00648	0.12953	0.54722	0.80201	0.91692
6.0	0.00278	0.09072	0.48691	0.76503	0.89862
6.5	0.00111	0.06137	0.42724	0.72482	0.87775
7.0	0.00041	0.04008	0.36949	0.68179	0.85425
7.5	0.00014	0.02525	0.31477	0.63644	0.82814
8.0	0.00004	0.01534	0.26404	0.58940	0.79952
8.5	0.00001	0.00898	0.21800	0.54138	0.76864
9.0	0.00000	0.00507	0.17710	0.49322	0.73590
9.5	0.00000	0.00275	0.14160	0.44591	0.70194
10.0	0.00000	0.00144	0.11154	0.40065	0.66775
10.5	0.00000	0.00072	0.08691	0.35904	0.63487
11.0	0.00000	0.00035	0.06782	0.32340	0.60563
11.5	0.00000	0.00018	0.05487	0.29733	0.58368
12.0	0.00000	0.00012	0.04982	0.28674	0.57463

Approaching Exascale: Best Practices for Training a Diverse Workforce using Hackathons

Izumi Barker
NVIDIA
Santa Clara, CA
ibarker@nvidia.com

Mozhgan Kabiri Chimeh
NVIDIA
Santa Clara, CA
mozhgank@nvidia.com

Kevin Gott
National Energy Research Scientific Computing Center
Berkeley, CA
kngott@lbl.gov

Thomas Papatheodore
Oak Ridge National Laboratory
Oak Ridge, TN
papatheodore@ornl.gov

Mary P. Thomas
University of California San Diego
La Jolla, CA
mpthomas@sdsu.edu

ABSTRACT

Given the anticipated growth of the high-performance computing market, HPC is challenged with expanding the size, diversity, and skill of its workforce while also addressing post-pandemic distributed workforce protocols and an ever-expanding ecosystem of architectures, accelerators, and software stacks. As we move toward exascale computing, training approaches need to address how to best prepare future computational scientists and enable established domain researchers to stay current and master tools needed for exascale architectures. This paper explores adding hybrid and virtual Hackathons to the training mix to bridge traditional programming curricula and hands-on skills needed among diverse communities. We outline current learning and development programs available; explain the benefits and challenges in implementing hackathons for training using experience gained from the Open Hackathons program (formerly the GPU Hackathons program); discuss how to engage diverse communities—from early career researchers to veteran scientists; and recommend best practices for implementing these events.

KEYWORDS

HPC, Exascale, Hackathons, HPC Training, HPC Education

1 INTRODUCTION

The potential for high-performance computing (HPC) to accelerate science is limitless, making it essential to much of the research activities across academia, supercomputing centers, government laboratories, and industry. As the landscape of research changes, large scientific projects can no longer advance in isolation but are dependent on community-driven participation. This necessitates the need for scalability of data processing and analysis, input/output capabilities that match pace with computational capabilities, and sufficiently performance-portable and expressive programming models that can handle the ever-growing volume, complexity, and rapidity of current and future data sets. [7]

The overall outlook for the HPC market is strong. Growing at an overall market compounded annual growth rate (CAGR) of 6.9 percent, Hyperion Research reported HPC spending (on-premise, cloud, and AI) for 2021 neared $35 billion (USD) and is on track to reach nearly $40 billion in 2022 and $50 billion by 2026. The rise of exascale and near-exascale systems has also seen tremendous growth, increasing from one near-exascale system in Japan in 2020 to five to eight exascale systems predicted by 2026 [9].

With the anticipated growth of HPC into exascale regions for both scientific computing and the broader enterprise, HPC is feeling the pressure of recruiting and retaining people. It faces the quandary of expanding the size, diversity, and skill of its workforce while simultaneously facing an expertise shortage. This scarcity of HPC experts is driven by several factors, such as the outflow of retirees exceeding the pipeline of new HPC staff, an increasing number of HPC sites worldwide, and the rising complexity of existing sites utilizing emerging technologies (i.e., AI, cloud, GPUs and other accelerators) that require different skill sets and leading to more systems per site [11].

As we move forward in exascale computing we must ask: How can we improve recruitment and better prepare future computational scientists for the upcoming challenges in exascale computing? How do we enable established domain researchers to stay current with the latest software and hardware trends and master the tools needed for the newer compute node architectures? How do we make exascale and HPC more accessible?

Traditionally, HPC has had a high barrier to use, owing in no small part to the shortfall of available expertise. Numerous training and development modalities exist, but often are independent of each other, lack standardization, or fail to incorporate real-world concepts and applications. Adding in-person and virtual hackathons to the training mix can bridge traditional programming curricula and hands-on skills needed among the diverse communities across national laboratories, supercomputing centers, and academic environments.

Hackathons and coding bootcamps have evolved from early coding and "bug discovery" sessions to become modern innovation events that combine agile programming and intense mentoring. The

https://doi.org/10.22369/issn.2153-4136/14/1/3

collaborative approach of these events provides the critical accelerated computing skills needed by the scientific community and the professionals that support them and aids in preparing researchers to use current and upcoming supercomputing resources.

This paper explores adding in-person and virtual hackathons to currently available learning and development programs, outlines the benefits and challenges of these events; discusses community impact and engagement, and finally, recommends best practices for implementing hackathons for ongoing and sustainable development.

2 BACKGROUND: CURRENT TRENDS IN HPC TRAINING AND EDUCATION TODAY

HPC concepts have been taught in academic settings, through informal webinars and tutorials at HPC Centers, or self-taught on an "as-needed" basis." In traditional academia, HPC content is interwoven within accredited computer science, information science, or computer engineering degree programs. For students pursuing research in other disciplines that require significant computing resources, HPC education may be integrated into courses in a student's subject domain (i.e., physics) but the number of institutions offering HPC coursework is low [14]. Additional challenges such as the diversity and complexity of the subject domains and the limited or varied computer literacy of the students only serve to compound the problem.

Modeling and simulation, now so ubiquitous, have led to emerging fields such as Computational Science and Engineering (CSE). Combining computer sciences, applied mathematics and statistics, and domain sciences, CSE's multidisciplinary approach encompasses methods of HPC and has become a cornerstone for the development and use of computational methods for scientific discovery [27]. While the number of CSE courses and programs has grown, the overall availability is low as is the number of students pursuing this area of study or graduating from these programs. Current coursework fails to expose students to real-world applications thus limiting a true understanding of the complexities of the field, preventing the development of skills needed for modern scientific and technological enterprises, and inadequately preparing students to fully utilize powerful new supercomputers for scientific applications and innovation. Moreover, almost no universities have a curriculum specifically focused on exascale or petascale science as issues are largely unknown and unexplored [3].

HPC education is also commonly taught as brief, condensed workshops lasting a half-day to several days or through specialized training modules and events. These workshops are offered by a diverse ecosystem of providers, but, whether it is an institution looking to shore up the skills of their existing staff, a government initiative aimed at ensuring the country continues critical research or a professional organization dedicated to a specific area of practice, the explicit goal is to develop a workforce with HPC-specific skills.

A variety of training options are available, ranging from webinars, lectures, Massively Open Online Courses (MOOCs) [15], hands-on labs and tutorials, software carpentries [29], on-the-job and specialized events among others. These options are often disparate, unrelated, and not universally standardized. Many of these training activities are executed in accordance with specific projects or agendas that may or may not continue, such as in the case of XSEDE which concluded formal operations as a National Science Foundation (NSF)-funded project in August 2022 [30], or the Exascale Computing Project (ECP), a component of the DOE-led Exascale Computing Initiative (ECI), which is moving to completion in mid-2023. An overview of the Training Efforts in the Exascale Computing Project can be found in the paper by Marques and Barker [12].

Additional challenges to workshops and training include a limited pool of qualified and available trainers, a finite number of workshops and training activities offered for a given region within the calendar year, the efficacy of the training materials and modalities to meet individual learning needs at scale, and finally, training materials may not be appropriate for the individual's specific project or timeframe.

3 LEVERAGING HACKATHONS FOR TRAINING

Complex scientific challenges and priority research is pushing the increased demand for extreme-scale computational resources to support a range of workflows for modeling, simulation, and data analysis that enable new discoveries and new understandings. The role of software that is reusable is central to research; however, many of the software libraries and scientific codes have largely been developed organically and maintained by a diverse community without considering longer-term sustainability that supports interdisciplinary collaboration nor addresses rapidly changing computing architectures [10].

Hackathons are fixed-time events during which individuals form teams and intensively collaborate to advance or complete a specific project of interest. [24] Believed to be coined during an OpenBSD cryptographic development event in 1999 [13], the meaning and nature of these events have developed from early ad-hoc exploratory programming sessions to represent modern innovation events that offer new opportunities for cooperative research and scientific discovery. Growing in both popularity and success, hackathons foster learning, drive community engagement, increase networking and relationship-building, and are effective for addressing civic, environmental, and public health issues, leading to increased adoption across various fields from higher education to healthcare to business services [8] [24].

For the purpose of this paper, we focus on Open Hackathons (formerly GPU Hackathons) [22], which are managed by the OpenACC Organization [23] and are designed to help scientists, researchers, and developers accelerate and optimize their applications on a variety of data center architectures, enabling them to build the critical skills needed to take advantage of modern HPC compute resources. Started as a one-off training activity in partnership with Oak Ridge Leadership Computing Facility [20] and NVIDIA [4] in 2014, Open Hackathons have evolved into a global program with over 100 hackathon events executed worldwide since the program launch.

3.1 Benefits

Leveraging Open Hackathons to support HPC training initiatives offers benefits to attendees, the hosting organizations, and the community at large. For hackathon attendees, the first and foremost benefit

is training and skills development. Where both academic settings and workshop/training offerings have challenges that perpetuate difficulties in meaningful HPC training at scale, Open Hackathons can address many of these limitations systematically.

At Open Hackathons, domain scientists are paired with experienced programming experts to receive dedicated guidance and mentorship for the course of the event. Attendees work with their mentors to strategically develop realistic goals for their codes and receive targeted recommendations and training in the HPC tools and resources relevant to those goals, allowing them to build hands-on skills such as learning how to compile their applications to identify computational bottlenecks or trying a new library or framework for a new approach to optimizing their code in short order.

Attendees are also given access to large heterogeneous HPC compute clusters that they may not otherwise have access to, for example, Ascent [21] the stand-alone 18 node system at OLCF with the same architecture and design as their Summit supercomputer, ranked in the top five of the Top500 list of most powerful commercially available computer systems since its debut in 2018 [25]. This enables an immersive experience that mirrors real-world environments so that attendees, particularly those who are students or early career researchers, can learn to navigate through many issues related to scalability, parallel efficiency, heterogeneous computing, parallel storage systems, and other issues [26]. Other computing platforms used during Open Hackathons include Cori at the National Energy Research Scientific Computing Center (NERSC), HiPerGator at the University of Florida, Juwels Booster at Forschungszentrum Juelich (FZJ), Piz Daint at the Swiss National Supercomputing Centre (CSCS), and Cirrus powered by EPCC [1] to name a few.

Since hackathon formats are intrinsically geared toward interdependent work, attendees benefit from collective knowledge sharing and increased opportunities for networking and recruitment as teams gain visibility to active projects and peers in different institutions and domains.

With the growing range of HPC workflows needing support and global exascale systems representing an $11 to $15 billion (USD) investment, it is imperative that HPC organizations compel full utilization of their existing systems and judiciously plan and prepare for upcoming needs. The motivations for organizations to host a hackathon are different from attendees; however, the benefits are aligned, focusing on skills development of their talent, system preparedness and utilization, talent recruitment, and competitiveness. To that end, Open Hackathons can assist.

Developing staff ability is paramount, since researchers cannot fully take advantage of computing systems without possessing the needed skills to do so. Hackathons help to facilitate quick and efficient skill-building through mentor engagement and guidance, hands-on team collaboration, and collective knowledge sharing. This is particularly effective, since the participants are actively working on their own specific codes or projects at the hackathon and therefore are deeply invested.

Many hackathons utilize the host organization's own compute cluster. This serves to assist staff in becoming more comfortable with the institution's available architectures and able to use new tools and techniques that allow them fully utilize the resource. Additionally, hackathons can help host institutions prepare for future system needs by giving them point-in-time snapshots of current research projects and their related applications across different domains of science, allowing the host to discover trends in the aggregate data and plan accordingly.

Lastly, as these events are most often open to the scientific community to participate in without regard to the participant's affiliation, hosting institutions can leverage hackathons to network, recruit new talent, as well as new users and projects of interest for computing allocation on their systems.

3.2 Open Hackathon Challenges

While there are numerous benefits to implementing Open Hackathons to augment HPC training, there are also challenges that need to be evaluated, including attendee preparedness, mentor availability and engagement, and system limitations.

Most Open Hackathon events are largely open to the scientific community for participation. This attracts a diverse applicant and attendee pool with varying levels of both domain-specific and technical skills and experience. This can lead to behaviors that affect participation, team dynamics, and overall outcomes. Applicants that are students or in early careers may feel intimidated or less able to fully contribute or participate without significant mentoring while more senior or seasoned attendees may be resistant to suggestions or new approaches. These behaviors affect team dynamics, impede progress and lead to lower satisfaction and learning outcomes.

Volunteer mentors are crucial to hackathon success: their expert guidance is needed to bridge gaps between domain knowledge and programming demands. With so many programming languages, libraries and frameworks, software development kits, and hardware choices available or utilized by the hackathon applications, having a large enough pool of qualified mentors can present a challenge when implementing a hackathon program. Given the intensive nature of hackathons, mentors must be available, engaged, and committed for the entirety of the event which can also pose challenges as they balance competing work priorities and schedules, particularly if they are affiliated outside the host organization.

Hackathons are great opportunities for training researchers on existing systems or helping to plan for upcoming system needs. They can also help stress test and evangelize systems that are newly online; however, this poses a challenge as well. A hackathon host should carefully consider the availability and "readiness" of the compute resource intended for the hackathon. Systems must be configured properly, be available and have adequate storage for the teams for the duration of the event, and provide any necessary information or instructions for access, containers, workflows and software stacks. Cluster support expectations should be understood and communicated. We have found that it's often best to have multiple systems available to mitigate risk associated with outages, new test systems, and other issues.

4 COMMUNITY IMPACT

The close integration of HPC simulation and data analysis continues to feed the development of new computer architectures and workflows, specialized software, and the growth of interdisciplinary teams, which are becoming more and more important for today's

HPC computing and scientific research efforts. Interdisciplinary research (IDR) is loosely defined as an effort conducted by teams that integrates information, data, techniques, tools, perspectives, and concepts from multiple disciplines to solve problems whose solutions are beyond the scope of a single discipline or area of research practice [19]. IDR is emerging to be a key concept of "convergence research," which is one of the NSF's "10 Big Ideas" for 2022 [6] and which the DOE Office of Science has made a Priority Research Area for SCGSR 2022. [18] Leveraging Open Hackathons as auxiliary HPC training programs will have a considerable, lasting impact on the research and development community by growing an interdisciplinary community, assisting with creating sustainable code, driving computer resource allocation, and advancing research. As a result, the Open Hackathon program has the potential to impact interdisciplinary scientific research and development.

The ethos of hackathons is collaboration and implementing these cooperative events for training grows the community by establishing wide-reaching, interconnected relationships between researchers and their projects. Participants can learn new perspectives, practices and technologies from each other, their mentors and their peers and are able to try new approaches in a safe environment. Strategic networks developed at these events can be instrumental for broader interdisciplinary knowledge exchange as well as raising the visibility of new collaboration opportunities and recruiting activities which is helpful for attracting new generations of HPC practitioners.

Scientific software is vital to research but faces difficulties in that it relies on an active community for continued development and distribution, but this community-driven approach can lead to an ecosystem of competing and collaborating products [10] as different contributors add to the codebase based on their own projects. Additionally, a sustainable approach to developing scientific software is sometimes overlooked by domain researchers as the focus is on publishing and not necessarily creating software [28]. As researchers work with mentors during the hackathon, not only is there a significant contribution to the code base but also an increased likelihood of a portable, production-ready, and sustainable code that can readily be used by the community since mentors are experienced programming experts and well-versed in creating reproducible, documented codes.

Open Hackathons increase community access to large-scale supercomputing systems enabling researchers and also act as feeders for additional initiatives and programs at hosting institutions as they solicit project proposals to allocate computing resources and cycles. One such example is the Innovative and Novel Computational Impact on Theory and Experiment, or INCITE program, jointly managed by Argonne Leadership Computing Facility and the Oak Ridge Leadership Computing Facility (OLCF) that awards allocations of supercomputer access to high-impact computational science projects across multiple disciplines [5]. Additionally, teams continuing work on large projects have participated in more than one hackathon, allowing them to access different compute systems (i.e., Ascent from ORNL [21] and Cori from NERSC [16]) aiding in scalability studies and comparisons.

Lastly, hackathons connect researchers to the right tools and technologies within an environment conducive to collaborative innovation and rapid optimization, making them very useful to advance research projects. To date, over 100 hackathons using this approach have been run worldwide and more than 550 scientific applications across multiple scientific domains have been accelerated wholly or in part at Hackathons. Examples include BerkeleyGW, Quantum ESPRESSO, CASTRO, Gkeyll, QUICK, CASTEP, and NWChem/NWChemEx. For additional information, please refer to the published paper: Best Practices in Running Collaborative GPU Hackathons: Advancing Scientific Applications with a Sustained Impact [2].

5 BEST PRACTICES FOR IMPLEMENTING HACKATHONS

Based on our experiences, we propose the following best practices in order to maximize the success and outcomes of hackathons and other training events.

5.1 Event Format

The Open Hackathon format centers around some guiding principles, including:

- detailed team application process involving hackathon hosts and Open Hackathon organizers to verify team capabilities and appropriate model to be studied,
- a minimum number of team members working on the same code to ensure a broader developer base behind the code [2],
- defined team goals for the hackathon [2], and finally,
- an approach that pairs teams of researchers with mentors and programming experts who are often experienced in the scientific domain.

An application process is utilized for participation in the hackathons where detailed information is collected, including code information such as programming language, programming model, algorithmic motif, code license, as well as team goals for hackathon and team members. Applications are reviewed by a jury composed of the host institution and program organizers in order to select those applications that 1) have a high impact project or code with domain relevance, 2) are technically feasible for the hackathon event with codes that are properly licensed, reproducible and documented, and 3) can be practically supported by available mentors in the network.

Most hackathon events run for a total of approximately five days; however, these days are not sequential but are separated over the course of two weeks to promote manageable and meaningful progress. “Day 0” occurs two weeks before the main hackathon event and introduces the team members and mentors, discusses the code, goals, and possible strategies to achieve these goals, and sets expectations between the participants. Day 0 also provides an overview of useful online tools as well as instructions for compute cluster access. “Day 1” occurs one week prior to the main event and introduces the participating teams, introduces all the mentors and their area of expertise, provides an overview of each project and code, and provides brief tutorials on the cluster, main tools such as profilers and libraries, and Q&A opportunities to encourage dialogue and knowledge sharing. The remainder of the hackathon (“Days 2-4”) occurs during the last week where teams and mentors work collaboratively, loosely applying agile methodology and presenting progress in daily stand-up scrum sessions.

Small adjustments can be made dependent on whether the event is in-person or virtual, but this format balances flexibility and discipline for optimal progress.

5.2 Team Composition and Preparedness

The guiding principles are coupled with careful team selection and preparation to make sure that teams make progress throughout the event and beyond.

In terms of team composition, we have found that three to five members is the ideal number for hackathon participation, permitting equitable division of work without too much "down" time. All team members should be fluent in the code they are working on and committed to completely participating for the duration of the hackathon event. Lastly, while team composition can vary greatly–from students to senior scientists, from little to no GPU or accelerator programming skills to advanced CUDA or language fluency–a balanced mix produces the best outcomes. For teams composed of all students, a principal investigator or advisor should be tasked with supervising and regular check-ins to keep goals aligned.

The more prepared the team is, the better the experience and ultimately, progress. There are many tools available to prepare for the hackathon and we recommend that teams take advantage of these to the extent possible. Focused training can shore up knowledge gaps and provide attendees with fundamental understanding of techniques that will be used during the hackathon. Targeted bootcamps are short-format training events that teach basic skills in specific topics (i.e., how to accelerate a code via various programming models) through a combination of lectures and hands-on activities using mini-applications in a controlled environment. These training events help attendees to quickly gain introductory programming skills that they can apply to a real code, increasing their confidence and readiness to participate in the hackathon. Finally, to maximize time with mentors, teams are encouraged to profile their codes ahead of time so that computational bottlenecks are known and can actively be addressed.

5.3 Team Mentors

Mentor pairing is one of the most critical components for the success of a team. Mentors should be assigned to teams based on several factors, including their core competencies, skill level, expertise and work style, and we recommend two mentors per team for most hackathons but this is flexible depending on the experience of the mentor and the complexity of the code/project.

Successful mentors have both the technical skills and soft skills. From a technical perspective, mentors are experienced programmers with core competencies in a specific programming language or model and who oftentimes also have domain-specific knowledge. This helps the mentor to understand the context of the problem statements and offer guidance that is specific to the goals of the team, and helps the team have confidence in the mentor and builds trust. Soft skills are also important to facilitating open communication, increasing receptiveness to coaching and keeping teams focused and on-track. For optimal outcomes, mentors should reinforce the focus on learning and development rather than "project completion," remaining in a mentoring role as opposed to project stakeholder. [17] This mentoring role mindset helps guide mentor interaction, informing mentors when to get "hands on" such as helping with small code samples, showing integration in the main code base, or profiling; and when to step back—allow team members to problem-solve or write code themselves to become self-sufficient.

Lastly, providing mentors with training and support helps increase success and satisfaction. Open Hackathons provides a variety of training options for mentors, including online courses, tutorials, industry training modules, peer-to-peer shadowing, and a mentor certification program.

6 CONCLUSIONS

High-performance computing (HPC) is critical to the continued advancement of science. As we approach the era of exascale computing, technology changes are creating opportunities and challenges, necessitating broadened approaches to training and developing the next generation of HPC users to be able to realize the full potential of emerging computing systems and architectures. Integrating hackathons into the training mix can bridge traditional programming curricula with real-world, hands-on skills to address the wide spectrum of computational needs and aptitudes and help stem the HPC talent shortfall.

ACKNOWLEDGMENTS

The authors would like to acknowledge Sunita Chandrasekaran, Guido Juckeland, Jack Wells, and Julia Levites for their continued leadership and support of the Open Hackathons program. Additionally, we would like to thank OpenACC-Standard.org, NVIDIA, and our extensive partner network for their support, including: Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725; National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231; Helmholtz-Zentrum Dresden Rossendorf, Jülich Forschungszentrum, Swiss National Supercomputing Centre; Brookhaven National Laboratory; the San Diego Supercomputer Center under NSF awards for Expanse (#1928224) and the Extreme Science and Engineering Discovery Environment(XSEDE) (#ACI-1548562), among others.

REFERENCES

[1] Edinburgh Parallel Computing Center. 2022. EPCC Cirrus System Page. Retrieved September 24, 2022 from https://www.cirrus.ac.uk/

[2] Sunita Chandrasekaran, Guido Juckeland, Meifeng Lin, Matthew Otten, Dirk Pleiter, John E. Stone, Juan Lucio-Vega, Michael Zingale, and Fernanda Foertter. 2018. Best Practices in Running Collaborative GPU Hackathons: Advancing Scientific Applications with a Sustained Impact. *Computing in Science & Engineering* 20, 4 (2018), 95–106. https://doi.org/10.1109/MCSE.2018.042781332

[3] Barbara Chapman, Henri Calandra, Silvia Crivelli, Jack Dongarra, Jeffrey Hittinger, Scott A. Lathrop, Vivek Sarkar, Eric Stahlberg, Jeffrey S. Vetter, and Dean Williams. 2014. DOE Advanced Scientific Advisory Committee (ASCAC): Workforce Subcommittee Letter. (7 2014). https://doi.org/10.2172/1222711

[4] NVIDIA Corporation. 2022. NVIDIA Home Page. Retrieved September 24, 2022 from https://www.nvidia.com/

[5] Department Of Energy. 2022. DOE INCITE Program. Retrieved September 24, 2022 from https://www.doeleadershipcomputing.org/

[6] National Science Foundation. 2022. Learn About Convergence Research. Retrieved September 24, 2022 from https://GOOGLE.COM

[7] Richard Gerber, James Hack, Katherine Riley, Katie Antypas, Richard Coffey, Eli Dart, Tjerk Straatsma, Jack Wells, Deborah Bard, Sudip Dosanjh, Inder Monga, Michael E. Papka, and Lauren Rotman. 2018. Crosscut report: Exascale Requirements Reviews, March 9–10, 2017 – Tysons Corner, Virginia. An Office of Science review sponsored by: Advanced Scientific Computing Research, Basic Energy Sciences, Biological and Environmental Research, Fusion Energy Sciences, High Energy Physics, Nuclear Physics. (1 2018). https://doi.org/10.2172/1417653

[8] Ahmed Imam, Tapajit Dey, Alexander Nolte, Audris Mockus, and James D. Herbsleb. 2021. The Secret Life of Hackathon Code Where does it come from and where does it go?. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. 68–79. https://doi.org/10.1109/MSR52588.2021.00020

[9] Hpyerion Research Joseph E. Sorensen. 2022. HPC Market Update Briefing During ISC22. Retrieved September 24, 2022 from https://hyperionresearch.com/hpc-market-update-briefing-during-isc22/

[10] Daniel S. Katz, Lois Curfman McInnes, David E. Bernholdt, Abigail Cabunoc Mayes, Neil P. Chue Hong, Jonah Duckles, Sandra Gesing, Michael A. Heroux, Simon Hettrick, Rafael C. Jimenez, Marlon Pierce, Belinda Weaver, and Nancy Wilkins-Diehr. 2019. Community Organizations: Changing the Culture in Which Research Software Is Developed and Sustained. *Computing in Science & Engineering* 21, 2 (2019), 8–24. https://doi.org/10.1109/MCSE.2018.2883051

[11] Hpyerion Research M. Riddle. 2022. ISC22 Market Update - HPC Talent Challenges. Retrieved September 24, 2022 from https://hyperionresearch.com/wp-content/uploads/2022/05/Hyperion-Research_ISC22-Market-Update_HPC-Talent-Challenges.pdf

[12] Osni Marques and Ashley Barker. 2020. Training Efforts in the Exascale Computing Project. *Computing in Science & Engineering* 22, 5 (2020), 103–107. https://doi.org/10.1109/MCSE.2020.3010596

[13] P. Mittai. 2022. A brief history of hackathon. Retrieved September 24, 2022 from https://content.techgig.com/codegladiators2021/a-brief-history-of-hackathon/articleshow/75291637.cms

[14] Julie Mullen. 2020. Interactivity, Engagement and Community Building in Online HPC Education and Training. Retrieved September 24, 2022 from https://sc20.supercomputing.org/proceedings/sotp/sotp_files/sotp124s2-file2.pdf

[15] Julia Mullen, Weronika Filinger, Lauren Milechin, and David Henty. 2019. The Impact of MOOC Methodology on the Scalability, Accessibility and Development of HPC Education and Training. *The Journal of Computational Science Education* 10, 1 (2019), 67–73. https://doi.org/10.22369/issn.2153-4136/10/1/11

[16] National Energy Research Scientific Computing Center (NERSC). 2022. NERSC Cori User Guide. Retrieved September 24, 2022 from https://www.nersc.gov/systems/cori/

[17] Alexander Nolte, Linda Bailey Hayden, and James D. Herbsleb. 2020. How to Support Newcomers in Scientific Hackathons - An Action Research Study on Expert Mentoring. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 25 (may 2020), 23 pages. https://doi.org/10.1145/3392830

[18] DOE Office of Science. 2022. Office of Science Priority Research Areas for SCGSR Program. Retrieved September 24, 2022 from https://science.osti.gov/wdts/scgsr/How-to-Apply/Priority-SC-Research-Areas

[19] National Academy of Science. 2005. Facilitating Interdisciplinary Research. Retrieved September 24, 2022 from https://www.doeleadershipcomputing.org/

[20] Oak Ridge Leadership Computing Facility (OLCF). 2022. Oak Ridge Leadership Computing Facility Home Page. Retrieved September 24, 2022 from https://docs.olcf.ornl.gov/systems/ascent_user_guide.html

[21] Oak Ridge Leadership Computing Facility (OLCF). 2022. OLCF Ascent User Guide. Retrieved September 24, 2022 from https://www.olcf.ornl.gov/

[22] OpenACC Organization. 2022. Open Hackathons Program. Retrieved September 24, 2022 from https://www.openhackathons.org/s/

[23] OpenACC Organization. 2022. OpenACC Organization Home Page. Retrieved September 24, 2022 from https://www.openacc.org/

[24] Ei Pa Pa Pe-Than, Alexander Nolte, Anna Filippova, Christian Bird, Steve Scallen, and James D. Herbsleb. 2019. Designing Corporate Hackathons With a Purpose: The Future of Software Development. *IEEE Software* 36, 1 (2019), 15–22. https://doi.org/10.1109/MS.2018.290110547

[25] Top 500 Project. 2022. Top 500 List. Retrieved September 24, 2022 from https://www.top500.org/lists/top500/

[26] Rajendra K. Raj, Carol J. Romanowski, John Impagliazzo, Sherif G. Aly, Brett A. Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I. Gordon, Cruz Izu, Shahram Rahimi, Michael P. Robson, and Neena Thota. 2020. High Performance Computing Education: Current Challenges and Future Directions. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 51–74. https://doi.org/10.1145/3437800.3439203

[27] Ulrich Rüde, Karen Willcox, Lois Curfman McInnes, and Hans De Sterck. 2018. Research and education in computational science and engineering. *SIAM Rev.* 60, 3 (2018), 707–754. https://doi.org/10.1137/16M1096840 arXiv:1610.02608

[28] Erik H. Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram, and James D. Herbsleb. 2014. Community Code Engagements: Summer of Code & Hackathons for Community Building in Scientific Software. In *Proceedings of the 18th International Conference on Supporting Group Work (GROUP '14)*. Association for Computing Machinery, New York, NY, USA, 111–121. https://doi.org/10.1145/2660398.2660420

[29] Greg Wilson. [n. d.]. Software Carpentry web site. http://software-carpentry.org. Main web site for Software Carpentry, replacing http://swc.scipy.org.

[30] XSEDE. 2022. XSEDE Project Home Page. Retrieved September 24, 2022 from https://www.xsede.org/

Teaching Accelerated Computing and Deep Learning at a Large-Scale with the NVIDIA Deep Learning Institute

Bálint Gyires-Tóth
Budapest University of Technology and Economics
Budapest, Hungary
toth.b@tmit.bme.hu

Işıl Öz
Izmir Institute of Technology
Izmir, Turkey
isiloz@iyte.edu.tr

Joe Bungo
Deep Learning Institute, NVIDIA Corporation
Austin, Texas
jbungo@nvidia.com

ABSTRACT

Researchers and developers in a variety of fields have benefited from the massively parallel processing paradigm. Numerous tasks are facilitated by the use of accelerated computing, such as graphics, simulations, visualisations, cryptography, data science, and machine learning. Over the past years, machine learning and in particular deep learning have received much attention. The development of such solutions requires a different level of expertise and insight than that required for traditional software engineering. Therefore, there is a need for novel approaches to teaching people about these topics. This paper outlines the primary challenges of accelerated computing and deep learning education, discusses the methodology and content of the NVIDIA Deep Learning Institute, presents the results of a quantitative survey conducted after full-day workshops, and demonstrates a sample adoption of DLI teaching kits for teaching heterogeneous parallel computing.

KEYWORDS

Accelerated Computing, Deep Learning, Artificial Intelligence, NVIDIA Deep Learning Institute

1 INTRODUCTION

Research and development have been transformed by the advancement of accelerated computing (AC). At present, the computational power of a single workstation is comparable to the power of a supercomputer of the past. Furthermore, the top supercomputer of today has broken the exascale barrier [23]. Due to the growing amount of data available, the significant enhancements in accelerated computing, and novel scientific results, deep learning (DL) [9] has become the most powerful tool for modeling real-world processes based on observations. In a neural network, the trainable parameters realized as a computational graph, are capable of learning various high- and low-level abstractions of the process being modeled, which is also referred to as feature learning. The modeling is performed hand in hand with the feature learning part in order to align the 'best' features with the 'best' model. Deep neural networks are scaling well – if more data is available, than a larger model can usually achieve better accuracy [6]. A robust hardware and software architecture for deep learning is capable of supporting the computational requirements. Aside from the ability to model speech [20] and vision [25] functions, deep learning is among the basic techniques for natural language processing [3], predictive maintenance [21], and anomaly detection [17], just to name a few areas. Professionals who are skilled in developing accelerated computing and deep learning solutions are in great demand. In these fields typically Pi or comb-shaped skills [10] are needed. A good understanding of fundamentals, programming skills, and project experience are essential even for a junior, which slows down the learning curve [8] compared to traditional education in software engineering. Besides higher education (HE), reskill [4] and upskill offerings of tech giants (like NVIDIA, Google, Amazon Web Service, Microsoft, etc.) and of vocational education training (VET) providers are among the possible options. Our paper discusses the main challenges in accelerated computing and deep learning education, demonstrates the methodology that was implemented in two universities based on the NVIDIA Deep Learning Institute (DLI) materials, and presents and discuss the results of the delivered contents.

2 EDUCATION

2.1 Accelerated Computing Education

Accelerated computing enables speed-up in program executions by leveraging hardware resources [5]. While instruction-level parallelism implemented in earlier superscalar processors provides performance optimizations and often does not need specific code modifications, leveraging multiple cores in a parallel system requires significant programming effort. Understanding the massively parallel execution and resource utilization in heterogeneous platforms with many-core GPUs requires expertise in architecture-aware programming.

While it is possible to introduce accelerated computing concepts in high-level directive-based programming models like OpenACC or OpenMP [2], teaching fine-grained programming based on low-level programming models like CUDA [7] or Pthreads can be an option to enable more parallelism opportunities for performance improvements in target executions.

For teaching heterogeneous computing, there are efforts to introduce parallel programming in different stages of undergraduate and graduate university education [18, 19]. Besides formal university courses, Massive Open Online Courses (MOOC)-style platforms enable people to learn about diverse topics by maintaining online

https://doi.org/10.22369/issn.2153-4136/14/1/4

courses. This solution seems promising as MOOC serves high-quality content from various qualified instructors and provides cloud infrastructure with software and hardware setup.

2.2 Deep Learning Education

Teaching deep learning can be approached in a variety of ways. Among the most common methods are:

Bottom-up: Generally, fundamentals such as probability theory, algebra, data analysis, and machine learning are taught first. Based on these concepts, backpropagation, stochastic gradient decent (SGD) and its variants, regularization techniques and traditional and modern neural architectures are described. Programming tasks and deep learning applications follow the fundamental components. Due to the fact that learning the fundamentals takes a considerable amount of time, this approach is usually taught in HE institutions as BSc and MSc programs. A combination of MOOC courses can also follow this approach.

Top-down: In order to gain practical experience as early as possible, the education begins with high-level programming examples. Following the first impression and the experience of success, participants are instructed on the fundamentals in greater detail. Depending on the length of the educational program, the depth of fundamentals may vary. In shorter courses, in MOOC courses, as well as in multi-semester programs for higher education, top-down approaches can be effectively incorporated.

Application-oriented: It is similar to the top-down approach, however it is geared towards a specific application domain, such as speech, computer vision, natural language processing, predictive maintenance, etc. Furthermore, the fundamentals are briefly discussed, mostly. Essentially, the goal is to gain knowledge about how to use DL tools in order to solve some specific problems. Application-oriented deep learning education are usually done in one to few-days trainings, workshops and boot camps.

Project-based [22] and on-the-job training: This focuses on some specific problem, which is often related to a real-world project. This approach allows corporate employees to gain deep learning experiences while working on their primary duties. In this case, not only the modeling but the data collection, preparation, feature engineering, and evaluation might be included in the training. In order to conduct a project-based or on-the-job training, senior deep learning experts are needed as instructors, who understand the problem, identify potential pitfalls, assist the employees in finding a solution (in which the expert is also involved), and evaluate that solution appropriately. A bootcamp or consultation service can be implemented using this approach.

3 METHODOLOGY

In this paper, we describe how NVIDIA Deep Learning Institute offerings help people to dive into AC and DL, and we also discuss, how these contents can be integrated into the academia. NVIDIA is a hardware and software platform company focusing on graphics processing units (GPUs) for the gaming and professional markets (including Artificial Intelligence), as well as system-on-a-chip units (SoCs) for the mobile computing and automotive market. Providing high quality software tools and educational materials is essential for NVIDIA in order to assist their customers. As for the former, it is provided by NVIDIA researchers and developers, while the latter is provided by NVIDIA Deep Learning Institute (DLI). NVIDIA DLI offers resources for diverse learning needs – from learning materials to self-paced and live training to educator programs—giving individuals, teams, organizations, educators, and students what they need to advance their knowledge in AI, accelerated computing, accelerated data science, graphics and simulation, and more. NVIDIA DLI has various offerings, as follows.

3.1 Self-Paced Courses

DLI offers online self-paced courses, where interested individuals follow the online materials from NVIDIA infrastructure on their own and receive certificates upon successful completion. Through accessing content on the latest technology trends prepared by experienced instructors and domain experts, and gaining hands-on experience with GPU-accelerated servers in the cloud, they learn to build deep learning, accelerated computing, and data science applications for a variety of industries. DLI offers self-paced courses in Deep Learning, Accelerated Computing Fundamentals, Data Science, Graphics and Simulation, Infrastructure, and Networking. The courses are in different lengths, from one- to eight-hours. Due to the various lengths, these courses are flexible to be integrated into university classes. For instance, after introducing the theory of Long Short-Term Memory (LSTM) in a bottom-up approach, including a DLI self-paced course on 'Modeling Time Series Data with Recurrent Neural Networks in Keras' [15] as a 2-hour-long practice helps students to have a hands-on experience with a real-world dataset. As the hardware and software infrastructure are already available, it is a great benefit to educators as well.

3.2 Instructor-led Workshops

For developers, data scientists, and engineers, live instructor-led workshops are taught by DLI-certified instructors with deep learning or accelerated computing expertise. The workshops may take place virtually or in-person with both models leveraging NVIDIA's online compute resources. Course materials include hands-on experience in a variety of concepts and levels. While some basic courses are instructor-led versions of the self-paced courses, there are many other advanced and domain-focused courses. By having a specific content, instructor-led workshops can be categorized as 'application-oriented' (see Section 2.2 for details). In addition to the actual applications, a broad theoretical overview is often presented as well, so the attendees can decide where to further their knowledge. DLI's instructor-led workshops cover five major areas:

Deep Learning Fundamentals teach how to use deep learning for computer vision, transformer-based natural language processing (NLP), conversational AI applications, recommendation systems, and multi-GPU setups.

Deep Learning by Industry describes how deep learning and AI can be applied to various industry domains such as industrial inspection, intelligent video analytics, anomaly detection, and predictive maintenance.

Accelerated Computing focuses on programming CUDA with C/C++ and Python on single and multiple nodes, as well as how to accelerate applications with OpenACC.

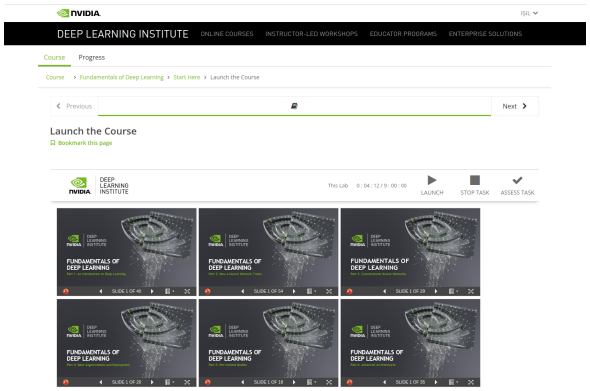


Figure 1: DLI workshop main page with slides and link to the cloud (via Launch Task).

Accelerated Data Science covers data science techniques accelerated with GPUs using Rapids.AI, and libraries such as cuDF, cuML, cuGraph, and more.

Networking introduces important concepts in building multi-GPU and multi-node systems.

Instructor-led workshops are offered by Deep Learning Institute for both individuals and teams from academia and industry. While public workshops are available for everyone, DLI University Ambassadors deliver free workshops for students and lecturers by utilizing hands-on course materials and GPU-accelerated workstations in the cloud. It is possible either to request a workshop from NVIDIA or to attend a scheduled workshop by registering for the course. Once registered for the offered workshop, an event code is sent to the participant via e-mail, and s/he can join the course from https://courses.nvidia.com/dashboard by creating an account in the system. After logging into the system, as seen in Figure 1, the participant can reach presentation slides, which the instructor explains during the workshop. Additionally, cloud-based GPU resources are available via Jupyter Notebook and JupyterLab interfaces. The participant can view both brief explanations and small examples, where he can execute code segments and modify them to get hands-on experience. In the meantime, he can access the workstation via the terminal to compile, execute, and modify the source files provided as part of the workshop. As the final part of the workshop, an assessment is given to demonstrate the information gained from the workshop and receive a certificate if the participant successfully completes the assessment. A typical assessment includes a hands-on programming goal, combining the main concepts taught in the workshop and testing the skills learned in the course. Moreover, some courses include only multiple-choice questions and require a minimum number of correct answers from the participant. While it is possible to attempt the assessment just at the end of the workshop, the participant can postpone the assessment evaluation and certification process. After completion of the workshop, the participants are asked to provide feedback about the workshop to evaluate both the content and the instructor.

The feedback form asks the following questions:

- How likely is it that you would recommend this course to a friend or colleague? (0..10)
- How would you rate these aspects of your learning experience? (1..5 and N/A)
  - Overall experience
  - Registration and login
  - Navigating the course
  - Launching hands-on content
- Did the course material meet your expectations? (1..5 and N/A)
  - Hands-on exercises were helpful in my learning objectives
  - Level of difficulty was as expected
  - Quality of content was as expected
  - The content of the course was interactive
  - Prerequisite information was useful
- How would you rate these aspects of your instructor-led session? (1..5 and N/A)
  - Instructor presentation skills
  - Instructor knowledge
  - TA knowledge
  - Pacing of course
  - Pre-event communication
- Anything else you'd like to tell us? (open ended question)

Teaching assistants (TAs) are involved depending on the number of participants. There should be one teaching assistant per 20 attendees as a general guideline. TAs are mainly helping in the chat. In case of a complex question, the TA will take the attendee into a breakout room for direct assistance. In this paper, we investigate the feedbacks of the following DLI workshops organized in Hungary by NVIDIA DLI and the Budapest University of Technology and Economics:

- Fundamentals of Deep Learning (FDL) [14]
- Building Transformer-Based Natural Language Processing Applications (NLP) [12]
- Building Conversational AI Applications (CAI) [11]

There were three different target groups (even within a group, the participants varied between two workshops):

- BSc group: These workshops were delivered as a part of a beginner level deep learning class (4 ECTS) at a university for BSc students.
- MSc group: The students were attending to a Human-Computer Interaction class (5 ECTS) at a university in their MSc studies.
- Mixed group: including BSc, MSc and PhD students, educators and non-profit researchers.

Participation in the workshop and passing the assessment were required for the BSc group to complete their course at the university. For the MSc group, passing the assessment was among the tasks to be exempted from the exam. Participants from mixed groups were invited to attend workshops (although it was not mandatory), and they were encouraged to pass the assessment to earn the certificate so they can add it to their CV. Participation in all workshops was free of charge, but only non-profit research institute and university staff and students were permitted to attend.

3.3 Teaching Kits

In order to assist educators in incorporating deep learning and accelerated computing into university courses, DLI offers downloadable

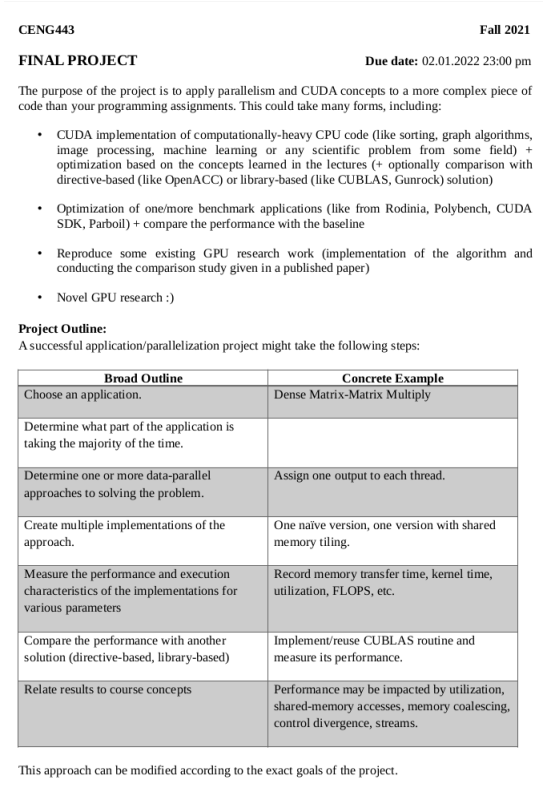
Table 1: Weekly Course Topics and Accelerated Computing Teaching Kit Modules.

Course Topic	Teaching Kit Module
Parallelism	Module 17 - Computational Thinking For Parallel Programming
Introduction to CUDA	Module 2 - Introduction to CUDA C
CUDA Threads	Module 3 - CUDA Parallelism Model
CUDA Memory	Module 4 - Memory and Data Locality
Tiling	Module 4 - Memory and Data Locality
Convolution	Module 8 - Parallel Computation Patterns (Stencil)
Parallel Patterns	Module 9 - Parallel Computation Patterns (Reduction) + Module 10 - Parallel Computation Patterns (Scan)
CUDA Performance	Module 6 - Memory Access Performance
Dynamic Parallelism	Module 23 - Dynamic Parallelism
CUDA Libraries	Module 25 - Using CUDA Libraries
CUDA CNN	–

teaching kits that include course materials that were co-developed with different university faculties. Each kit, freely available for the instructors world-wide, includes lecture slides and hand-on lab exercises with sample solutions. Additionally, the Teaching Kits Program provides free access for instructors and students to GPU-accelerated workstations in the cloud, either through Amazon's AWS program offering credits or online self-paced DLI courses. (mentioned in Section 3.1). The students can access GPU resources for hands-on exercises or larger-scale projects, and earn certificates that demonstrate their expertise in the subjects.

In the computer engineering department at Izmir Institute of Technology in Turkey, the Heterogeneous Parallel Programming course has been offered based on the Accelerated Computing teaching kit. The semester-long technical elective course covers GPU hardware, CUDA basics, advanced CUDA features, and parallel application development topics. While the content is updated each year, the main concepts and the corresponding teaching kit modules are presented in Table 1.

While the slides from the teaching kit are utilized in the specific modules, lab exercises and quiz questions are not used since there is no lab session or quiz in the course. Instead, self-developed programming assignments and midterm/final questions are designed for the course assessment and evaluation. Additionally, a final term project is assigned to the students, where *Project Guidelines* document of the Teaching Kit is utilized for defining the purpose, outline, and grading rubric of the project (The definition document at 2020-2021 term is given in Figure 2). The students are expected to propose and implement a complete CUDA application, conduct an experimental study, and perform a comparative analysis by comparing different CUDA implementations with other programming models, like OpenACC or other libraries.



CENG443 **Fall 2021**

FINAL PROJECT **Due date:** 02.01.2022 23:00 pm

The purpose of the project is to apply parallelism and CUDA concepts to a more complex piece of code than your programming assignments. This could take many forms, including:

- CUDA implementation of computationally-heavy CPU code (like sorting, graph algorithms, image processing, machine learning or any scientific problem from some field) + optimization based on the concepts learned in the lectures (+ optionally comparison with directive-based (like OpenACC) or library-based (like CUBLAS, Gunrock) solution)
- Optimization of one/more benchmark applications (like from Rodinia, Polybench, CUDA SDK, Parboil) + compare the performance with the baseline
- Reproduce some existing GPU research work (implementation of the algorithm and conducting the comparison study given in a published paper)
- Novel GPU research :)

Project Outline:
A successful application/parallelization project might take the following steps:

Broad Outline	Concrete Example
Choose an application.	Dense Matrix-Matrix Multiply
Determine what part of the application is taking the majority of the time.	
Determine one or more data-parallel approaches to solving the problem.	Assign one output to each thread.
Create multiple implementations of the approach.	One naïve version, one version with shared memory tiling.
Measure the performance and execution characteristics of the implementations for various parameters	Record memory transfer time, kernel time, utilization, FLOPS, etc.
Compare the performance with another solution (directive-based, library-based)	Implement/reuse CUBLAS routine and measure its performance.
Relate results to course concepts	Performance may be impacted by utilization, shared-memory accesses, memory coalescing, control divergence, streams.

This approach can be modified according to the exact goals of the project.

Figure 2: Final Term Project Definition at the Heterogeneous Parallel Programming Course.

3.4 Hardware and software infrastructure

In order to conduct research, development, and education in AC and DL, a specific hardware and software infrastructure is required. In terms of hardware, the most critical component is access to GPU(s), since they are not commonly found in personal computers. Further, the appropriate software stack is required, which includes drivers for the GPU(s) and the programming environment, frameworks, and modules relevant to the topic. Integrated development environments (IDEs) should also be easily accessible to users. Setting up an appropriate hardware and software environment for AC and DL education can be time-consuming and costly. Since one of the main goals of DLI courses is to provide hands-on programming exercises that are to be executed on GPU-based architectures, NVIDIA provides access to the participants NVIDIA GPU enabled cloud environment with all necessary software components installed. The software stack is built in separate Docker images [1], and the IDE

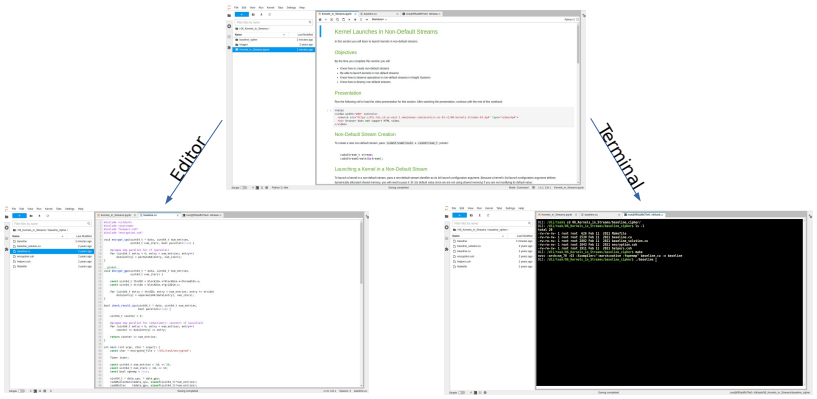


Figure 3: Sample module interfaces in Fundamentals of Accelerated Computing workshop.

is primarily a web-based application (Jupyter Notebook and Lab, https://jupyter.org/).

The participant can utilize the cloud resources presented as Jupyter notebooks, which can be accessed by graphical notebook interface, graphical console IDE, or terminal screen. While graphical interfaces are more useful for Python-based courses like Fundamentals of Deep Learning, terminal provides more practical interface like Fundamentals of Accelerated Computing, which may require frequent source code modification and low-level analysis. Figure 3 presents one module (*Kernels_In_Streams*) and possible user interfaces in *Fundamentals of Accelerated Computing* workshop to access the module components. While the main Jupyter Notebook interface provides guidance about the module, the participant can edit the source code in the editor interface or modify/compile/execute in a terminal screen. Additionally, the courses that include visual performance analysis, based on NVIDIA Nsight Systems tool [16], offer remote desktop access, which has running Nsight Systems instance inside the JupyterLab environment. The participants can connect this desktop environment and visually profile their executions by observing performance behavior of the different code versions to see the effects on performance. Figure 4 demonstrates the phases for using remote Nsight Systems tool in DLI infrastructure:

(1) Executing the program in the terminal with *profile* option (provided in Makefile),
(2) Connecting the remote desktop and observing the report file generated at the end of the program execution,
(3) Visualizing the profile report at Nsight Systems Tool, which is already installed and configured in the remote desktop environment.

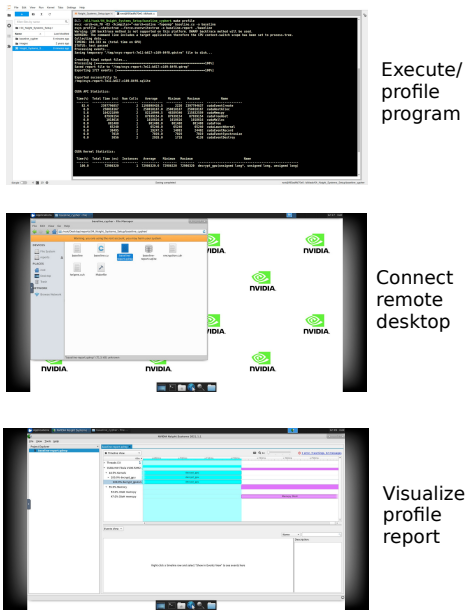


Figure 4: Nsight Systems Tool in remote desktop.

3.5 University Ambassador program

The DLI University Ambassador Program [13] enables qualified educators to teach free instructor-led courses for the academia, including university and non-profit research lab staff, students, and researchers. They are also allowed to run paid corporate workshops. By completing the instructor certification process, educators affiliated with an academic institution are certified as University Ambassadors. For each workshop, DLI instructors must pass a multi-stage examination in order to become certified in the specific content. Teaching assistants are selected by the instructors. This program has several benefits: free DLI instructor certification, online ready-made workshop materials, free access to online GPU resources, and expense reimbursement for travel and catering expenses for instructor-led workshops. See [13] for detailed information about this program.

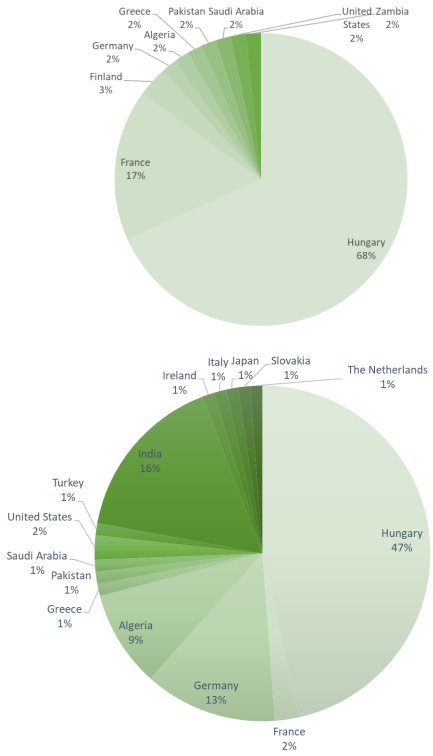


Figure 5: Country of origin of the attendees in the mixed group deliveries, on the top: FDL with 60, on the bottom: NLP with 86 attendees.

4 EVALUATION AND RESULTS

4.1 Instructor-led Deep Learning Workshops

Altogether we held 2 FDL, 1 NLP, and 3 CAI workshops in the autumn and spring semesters of 2021/2022 academic year, according to Section 3.2. All of these workshops were ran by an associate professor with 10+ years of machine learning, and 8+ years of deep learning research, development, and education experience. The number of participants of the examined workshops was as follows:

- BSc group: one FDL and one CAI were delivered in-class for 30 (22 from Hungary, 7 from the USA, 1 unknown) and 26 (22 from Hungary, 4 from the USA) students, respectively. These workshops were delivered as a part of a beginner level deep learning class at a Hungarian university.
- MSc group: two CAI were delivered online for 13 (12 from Hungary, 1 from Romania) and 38 (37 from Hungary, 1 from the USA) attendees. The students were attending to a Human-Computer Interaction class at a Hungarian university.
- Mixed group: one FDL and one NLP were delivered online for 60 and 86 attendees, respectively. The attendees' country of origin are shown in Fig. 5. These workshops were advertised in various channels in the EMEA region, including AI-related mailing list in Hungary, LinkedIn groups, and NVIDIA DLI academic partners.

Figure 6, 7, 8 show the results of the feedback forms.

Learning experience. The overall impression of the attendees was 4 or above. A weak but clear trend can be inspected that the more knowledgeable the audience was, the higher they scored the overall experience (4 and a little bit below for the BSc, 4 and a little bit above for the MSc, and around 4.5 for the Mixed group). Interestingly, similar trend is shown for the other questions (Registration, Navigation, Launch Time), however, those aspects are not directly correlated to hard skills, knowledge, and experience. There are two possible explanations for this. On the one hand, juniors are more likely to get frustrated than senior experts. There were more seniors in Mixed than in BSc and MSc groups, since it included PhD students, researchers, and educators in addition to BSc and MSc students. On the other hand, participants of mixed groups were attending the workshop on their own initiative and during their free time, so they recognized the value of the material more than university students, for whom the content was part of their course work.

Meeting the expectations. In all groups, meeting the learning objectives scored 4 or above – with the Mixed group scoring the highest. In spite of having different groups and different contents, the difficulty of the materials was considered to be similar. It reinforces that NVIDIA DLI's efforts to maintain a dense information content in the courses, but in a manner that is digestible in a full-day workshop are successful. Similar scores can be inspected for the 'clear prerequisites'. The quality of the content was scored better by more advanced groups (MSc and Mixed), and it scored 4 for the BSc group, too. In interactivity, similar weak trend can be inspected, as before. It is interesting that within the same groups FDL scored higher than the more advanced NLP and CAI content, regarding interactivity. This can be mainly the cause of the course content: When introducing deep learning for the first time, more interactions are involved in the workshop. When discussing advanced topics like NLP or CAI, the participants are considered to be more advanced, thus information content is superior to interaction.

Instructor, teaching assistants, course pace. Feedback about the instructor showed similar patterns as the previous two categories. The instructor's presentation skills and knowledge were judged quite similar by distinct groups. Interestingly, among all questions the feedback on the teaching assistant's (TA's) knowledge scored the lowest overall. The workshop TAs were all PhD candidates specialized in deep learning, they had teaching and consultation experience, and they had earned the certificate of the particular workshop in advance. The relatively lower scores (<4) may be the result of different expectations of the TAs (e.g. expecting more help in the self-paced parts of the workshop) and/or the way TAs interacted with the audience degraded the participants' experience (chat, generally).

The statistics of successful certificates are shown in Table 2. Due to the requirement to earn the certificate in order to complete the deep learning course at the university, it is understandable why the majority of attendees completed the assessment successfully in the BSc groups. In case of the MSc groups a smaller percentage of the class earned the certificate – in this case the certificate was not required, but was among the options to be exempted from the exam. In the case of the Mixed-FDL similar percentage of the group passed the assessment successfully. For Mixed-NLP the percentage dropped significantly, to 44%. The possible cause for this could be

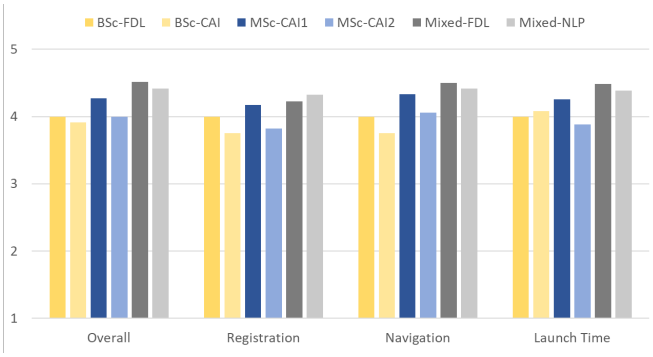


Figure 6: Results of the feedback form on the learning experience.

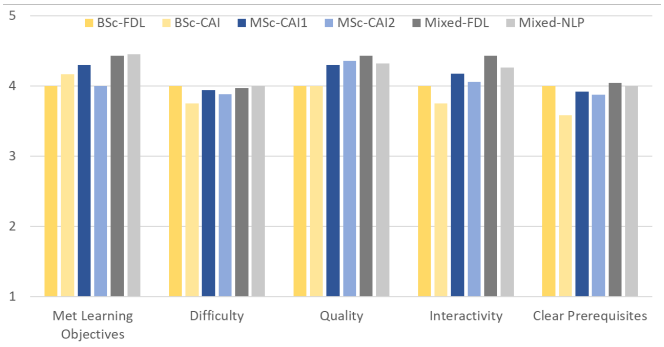


Figure 7: Results of the feedback form on the course meeting the expectations.

the timing of the workshop: this one was held in 13 December, right before the holiday season, when students and educators are also overloaded with exams, and researchers with finalizing projects at the end of the year – which allow them less time to completely participate in a full day workshop and complete its assessment.

Table 2: Percentage of participants who have obtained a certificate by completing the assessment in the given workshop.

Workshop	Percentage
BSc-FDL	94%
BSc-CAI	93%
MSc-CAI1	77%
MSc-CAI2	64%
Mixed-FDL	69%
Mixed-NLP	44%

4.2 Adopting Accelerated Computing Teaching Kit

Each year 10-20 students are registered in the Heterogeneous Parallel Programming course, and in average 60-80% of them can get

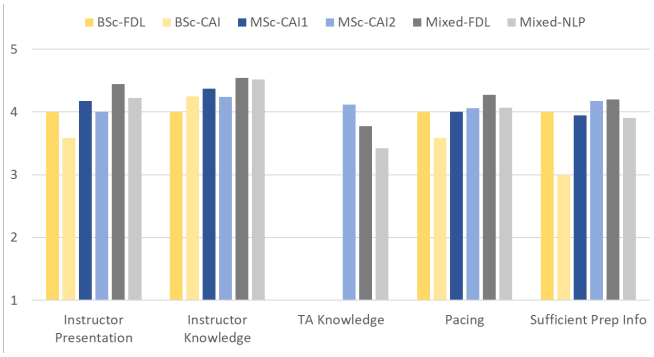


Figure 8: Results of the feedback form on instructor, teaching assistant and course pace (in BSc-FDL, BSc-CAI and MSc-CAI1 there were no teaching assistant.

a passing grade. Table 3 presents the statistics about the course in the four years. It presents the number of students in terms of enrolled in the course, failed (got F) from the course, and received the highest letter grade, AA. Additionally, *Course Evaluation* column demonstrates the average score of the evaluation survey (out of 5), where the number in parenthesis represents the score for the question about the demonstration of the course content based on the quality of the course material and effective examples. While

Table 3: Heterogeneous Parallel Programming course statistics.

Term	#Students Taken	#Students Failed	#Students w/ AA	Course Evaluation
2021-2022	11	2	1	4.23 (4.27)
2020-2021	12	4	5	2.84 (2.89)
2019-2020	20	8	3	3.94 (3.94)
2018-2019	16	6	1	3.79 (3.60)

general feedback appreciates the effort in the course, term *2020-2021* demonstrates a negatively different result with relatively low scores in the course evaluation. Since the course is taught virtually that term, we think that student involvement could not be achieved as in the face-to-face semesters. It is also remarkable that *2021-2022* evaluation results are the highest even though the number of students is not large. Since *CUDA Libraries* and *CUDA CNN* are emphasized that year, we think that the students were able to see the power of CUDA programming model and real scenarios that they can apply the methods and, as a result evaluated the course as more efficient.

For the student evaluation, programming tasks were assigned to the students to demonstrate their comprehension of the concepts introduced throughout the semester. Additionally, the final project tests their skills at defining parallel programming problems, optimizing performance by considering GPU hardware and CUDA programming model features, and performing a comparison study to evaluate the effectiveness of their methods. The sample project topics in 2021-2022 semester were as follows: Perlin and fractal noise, Gaussian Jordan elimination, Dijkstra's shortest path algorithm, Convolution operations from the PolyBench benchmark. The

Figure 9: Number of additions and deletions per week in the GitHub projects.

students created GitHub repositories and updated their code during the semester based on a few deadlines. Figure 9 presents the code frequency in terms of additions and deletions in sample GitHub projects. In the two-month period, there are peaks at two specific points representing the deadlines. While most of the projects include basic CUDA implementations, one project is extended as a conference paper and presented at a national conference by the student [24].

5 SUMMARY

In this paper the primary challenges of accelerated computing and deep learning education was introduced, the offerings of NVIDIA Deep Learning Institute were discussed and instructor-led full day workshops and teaching kits were evaluated. The feedback form filled after the workshops revealed that in case of all examined content the overall satisfaction with the learning experience were between 3.9...4.5 (out of 5). The results also showed us, that more experienced groups scored various aspects higher (e.g. overall impression, quality of the content, interactivity, impressions about the instructor, etc.). No significant difference in difficulty was observed between beginner and advanced workshops, based on the feedback scores. Surprisingly, experienced teaching assistants received rather lower scores (between 3.4..4.3) compared to other questions in the feedback forms.

Based on the course evaluation questions and the implementation of the term projects, we can conclude that the adoption of Teaching Kits was a success.

It is our overall impression and conclusion that the content created by NVIDIA DLI can be easily and successfully integrated into related university courses for smaller and larger groups. DLI content can even be implemented in classes that are not directly related to AC or DL (e.g. the Human-Computer Interaction MSc course) with a great learning experience – based on our findings.

ACKNOWLEDGMENTS

The authors are thankful for the support of NVIDIA Deep Learning Institute. The work reported in this paper has been partly supported by the the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory.

REFERENCES

[1] Carl Boettiger. 2015. An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review* 49, 1 (2015), 71–79.
[2] Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. 2001. *Parallel programming in OpenMP*. Morgan kaufmann.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[4] Rebecca Fiebrink. 2019. Machine learning education for artists, musicians, and other creative practitioners. *ACM Transactions on Computing Education (TOCE)* 19, 4 (2019), 1–32.
[5] John L Hennessy and David A Patterson. 2011. *Computer architecture: a quantitative approach*. Elsevier.
[6] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409* (2017).
[7] David B Kirk and W Hwu Wen-Mei. 2016. *Programming massively parallel processors: a hands-on approach*. Morgan kaufmann.
[8] Karen Kreeger. 2003. The learning curve. *Nature Biotechnology* 21, 8 (2003), 951–952.
[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
[10] Linda Macaulay, Claire Moxham, Barbara Jones, and Ian Miles. 2010. Innovation and Skills. In *Handbook of Service Science*. Springer, 717–736.
[11] NVIDIA. 2022. Building Conversational AI Applications. Retrieved September 8, 2022 from https://www.nvidia.com/en-us/training/instructor-led-workshops/building-conversational-ai-apps/
[12] NVIDIA. 2022. Building Transformer-Based Natural Language Processing Applications. Retrieved September 8, 2022 from https://www.nvidia.com/en-us/training/instructor-led-workshops/natural-language-processing/
[13] NVIDIA. 2022. DLI University Ambassador Program. Retrieved September 8, 2022 from https://www.nvidia.com/en-us/training/educator-programs/university-ambassador-program/
[14] NVIDIA. 2022. Fundamentals of Deep Learning. Retrieved September 8, 2022 from https://www.nvidia.com/en-us/training/instructor-led-workshops/fundamentals-of-deep-learning/
[15] NVIDIA. 2022. Modeling Time Series Data with Recurrent Neural Networks in Keras. Retrieved September 8, 2022 from https://courses.nvidia.com/courses/course-v1:DLI+L-FX-24+V1/
[16] NVIDIA. 2022. NVIDIA Nsight Systems. Retrieved September 8, 2022 from https://developer.nvidia.com/nsight-systems
[17] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.
[18] Apan Qasem and David P. Bunde. 2022. Heterogeneous Computing for Undergraduates: Introducing the ToUCH Module Repository. In *SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education, Providence, RI, USA, March 3-5, 2022, Volume 2*. ACM, 1201. https://doi.org/10.1145/3478432.3499152
[19] Apan Qasem, David P. Bunde, and Philip Schielke. 2021. A module-based introduction to heterogeneous computing in core courses. *J. Parallel Distributed Computing* 158 (2021), 56–66. https://doi.org/10.1016/j.jpdc.2021.07.011
[20] Samik Sadhu, Di He, Che-Wei Huang, Sri Harish Mallidi, Minhua Wu, Ariya Rastrow, Andreas Stolcke, Jasha Droppo, and Roland Maas. 2021. Wav2vec: A self-supervised model for speech representation learning. *arXiv preprint arXiv:2103.08393* (2021).
[21] Oscar Serradilla, Ekhi Zugasti, Jon Rodriguez, and Urko Zurutuza. 2022. Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. *Applied Intelligence* (2022), 1–31.
[22] Daniel Spikol, Emanuele Ruffaldi, Giacomo Dabisias, and Mutlu Cukurova. 2018. Supervised machine learning in multimodal learning analytics for estimating success in project-based learning. *Journal of Computer Assisted Learning* 34, 4 (2018), 366–377.
[23] Top500.org. 2022. ORNL's Frontier First to Break the Exaflop Ceiling. Retrieved August 11, 2022 from https://www.top500.org/news/ornls-frontier-first-to-break-the-exaflop-ceiling
[24] Burak Topçu and Işıl Öz. 2022. Performance Evaluation of CUDA Optimizations for Convolution Operations. In *Yüksek Başarımlı Hesaplama Konferansı (BAŞARIM)*. https://indico.truba.gov.tr/event/50/attachments/231/457/BASARIM2022_Proceedings.pdf
[25] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*. PMLR, 23965–23998.

Preliminary Results of Applying Modified MSA Algorithm on Quantum Annealers (MAQ)

Melody Lee
The North Carolina School of Science and Mathematics
Durham, NC
lee23m@ncssm.edu

ABSTRACT

We propose a modified MSA algorithm on quantum annealers with applications in areas of bioinformatics and genetic sequencing. To understand the human genome, researchers compare extensive sets of these genetic sequences – or their protein counterparts – to identify patterns. This comparison begins with the alignment of the set of (multiple) sequences. However, this alignment problem is considered nondeterministically-polynomial time complete and, thus, current classical algorithms at best rely on brute force or heuristic methods to find solutions. Quantum annealing algorithms are able to bypass this need for sheer brute force due to their use of quantum mechanical properties. However, due to the novelty of these algorithms, many are rudimentary in nature and limited by hardware restrictions. We apply progressive alignment techniques to modify annealing algorithms, achieving a linear reduction in spin usage whilst introducing more complex heuristics to the algorithm. This opens the door for further exploration into quantum computing-based bioinformatics, potentially allowing for a deeper understanding of disease detection and monitoring.

KEYWORDS

Quantum Annealing, Multiple Sequence Alignment, Bioinformatics, Clustering, Progressive Alignment, Spin Use Reduction

1 INTRODUCTION

1.1 Alignments in Disease Detection and Prevention

In a single year, over 850 million years of healthy life may be lost to disease and disabilities [33]. In fact, an estimated 50% of the United States population is living with a chronic disease [15]. Presently, there is an inadequate response to this healthcare crisis, as most of the attention in epidemiological research and health care has been centered around acute diseases [15]. Human genomes are explicit factors in determining susceptibility to some of these diseases [22]. Comparison of their constituent genetic sequences may one day reveal knowledge that permits for early diagnosis or monitoring of heritable diseases of at-risk individuals [17]. Furthermore, the comparison of sequences has heavy bearing on treatment procedures as well. For instance, the study of large sets of DNA sequences can allow researchers to eventually predict patient response to chemotherapy or other treatments [32]. This could allow for the determination of personalized treatment options for patients in order to maximize their chances of recovery, including cancer. Therefore, emphasis on the comparison of the genetics underlying major actors in these diseases – from protein mutations to patient genomes – is needed.

There is a clear issue, however. The sheer length of genetic sequences is comparable to the circumference of the Earth or even the distance to the moon. Each genetic sequence can contain on the magnitude of several thousand base pairs. Analysis of large sets of these sequences consumes significant computing resources. Protein sequences are no better, with the length of the amino acid sequences on a similar order of magnitude. In spite of the limited alphabet these sequences are composed of – pulling from sets of a mere four base pairs or twenty amino acids – these sequences are responsible for the behavior of countless diseases in existence, and thus researchers have sought various methods of analyzing them.

Table 1: An example alignment for a set of three genetic sequences.

A	T	G	-	T	T
A	T	-	C	T	T
T	T	G	C	T	-

To compare these sequences effectively, an ideal alignment of the sequences must be found, in which gaps or shifts in the sequences are inserted to minimize the differences in each column of Table 1. After all, it would do no good if subsequences that encode for different biological components are mistakenly compared against one another. The problem of finding the multiple sequence alignment (MSA) is an applied form of the mathematical consensus string problem [34]. The solution seeks to find an alignment where the distances between sequences are minimized. This distance is a quantitative measurement of how well sequences are aligned, comparable to the aforementioned number of differences in each column [16]. For every alignment of a pair of sequences, the elements in corresponding positions are compared. The greater the discrepancies across the positions, the greater the distance between the two sequences [16]. This problem is analogous to finding the smallest distance between some set of locations. The given set of locations are the sequences, and their distances are the differences between each plausible alignment. The nature of this problem, therefore, centers on distance minimization, deeming it an optimization problem. While the alignment of, say, ten or twenty elements per sequence is not difficult, solving the problem for larger and larger scales can become unmanageable for the standard human mind.

https://doi.org/10.22369/issn.2153-4136/14/1/5

This paper functions as a simultaneous investigation into MSA algorithms and certain alterations to these algorithms that may be made. We begin by discussing existing algorithms for MSA, alongside shortcomings in the computational tools currently in use. We then transition to discussion of quantum annealing, prior to discussing our classical-inspired modifications to a MSA quantum annealing algorithm. This modified algorithm is then used to align a sample dataset and its results analyzed.

1.2 Existing Algorithms for MSA

Rather than arbitrarily align these sequences, MSA algorithms systematically align sequences. While there are numerous algorithms in existence, the most common are based on the Needleman-Wunsch algorithm. This algorithm was first described in 1970 by its namesake researchers, Saul B. Needleman and Christian D. Wunsch [29].The initial algorithm aligns a pair of protein sequences using iterative comparison of each individual amino acid [29]. While effective, there lies a major issue in the resource requirements for the problem. The MSA problem has non-deterministic polynomial-time hardness (NP-hardness) [34]. As the size of the input size increases, the amount of time and memory required to find the perfect solution increases at unmanageable rates. This becomes a hindrance in effective application. Finding the ideal alignment for inputs on the same magnitude as protein or genetic sequences can take decades to process. Thus, better algorithms capable of handling larger inputs are being sought.

Over the years, researchers have developed more complex methods of computation to raise the ceiling on the size of the inputs that may be reasonably handled. Algorithms capable of running on multiple computer cores in parallel have been developed. This is analogous to having multiple people brainstorm ideas for a project, as opposed to a singular "brain" working on the task. The approach has approximately a 60% reduction in execution time from experimental results, showing parallel processing has strong potential [28].

Another common – but effective – method aligns smaller subsets of the sequences before merging the final solution. These methods are generally categorized into two types: (1) progressive alignments and (2) iterative alignments [12]. Progressive algorithms organize sequences based on similarity and arrange subsets of these sequences. In some cases, the sequences are arranged in a tree-like structure, such that only a few sets of parents and their children are aligned at once, reducing the load on the computer at any single point [12]. Iterative algorithms, on the other hand, go through multiple iterations of aligning and then re-aligning sequences in overlapping subsets. Both types may use heuristics to estimate the pairwise distances between sequences prior to arrangement, allowing them to introduce reasonable steps that increase the scalability of the resultant process [42].

1.3 A Tool for Problem Solving: Quantum Computers

While existing methods are effective effective, they are still bound by the binary nature of computing units. That is, standard classical computers have bit values restricted to either 0 and 1, or True and False, and therefore are only able to represent one state at a time

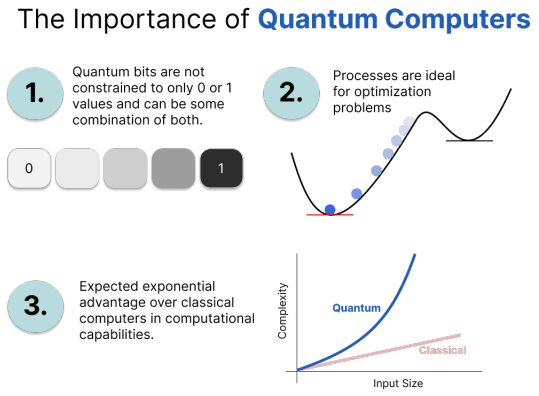


Figure 1: There exist several key characteristics of quantum computers that make them especially of interest when it comes to algorithms (created by author).

[35]. Quantum computers – which make use of parallel processing and quantum mechanical properties to bypass these restrictions – have emerged as new contenders for finding alignments [44].

While the absolute supremacy of quantum computers over their classical counterparts is yet unproven [36], they have two key properties whose partnership make computation on quantum systems especially advantageous: one, parallel processing and, two, entanglement. The parallel processing capabilities come from the ability for the quantum bits to be in a probabilistic suspension between the bit values, or in a superimposed state [44]. This phenomenon allows for an exponential number of solutions to be simultaneously represented [35]. This cooperates with the second property, entanglement, to make quantum computers especially unique. The values of the quantum bits – including those in superposition – may be "tangled" together, such that knowledge of the value of one qubit will reveal information about other entangled qubits in the system [35]. This permits for added levels of complexity [35]. The combination of these quantum mechanical properties in computing makes quantum computing especially well-suited for solving NP-hard problems (Figure 1).

For example, certain algorithms have used a combination of both classical techniques and quantum computer capabilities. Researchers have applied machine learning models to reduce the amount of memory required to store comparisons of the sequences [40]. Others have taken inspiration from the quantum mechanical properties outright in developing quantum-inspired heuristics to find alignments [12].

1.3.1 Quantum Annealing Algorithms. Other algorithms focus on a subtype of quantum computing: quantum annealing. Quantum annealers, also known as adiabatic quantum computers, take advantage of the natural tendency for physical systems to seek out the lowest energy configurations [8]. A commonly used analogy to illustrate the workings of a quantum annealer involves finding the lowest point of elevation among a series of hills and valleys [8]. This region is analogous to the problem space defined. Classical computers find the solution to the problem by sending a singular traveler to begin at some arbitrary point in the area. This traveler finds the

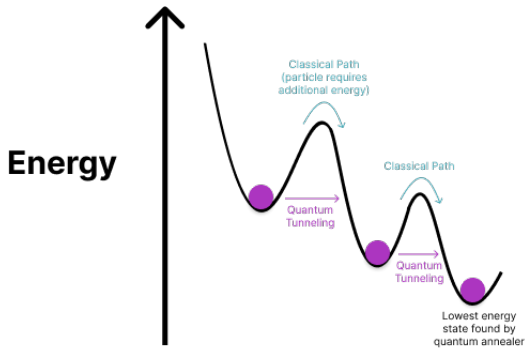


Figure 2: Quantum annealers use quantum tunneling to find the lowest energy state for the given problem space (created by author).

minimum by walking some direction determined by the classical algorithm until a local lowest point is reached. To ensure this is the absolute lowest point, the classical algorithm then proceeds to drop the traveler off again at several other locations across the area. Quantum annealers, on the other hand, bypass this repeated traversal. Rather, superposition permits for the traveler to exist simultaneously in different locations, cutting down significantly on the costliness of traversal [8]. To find the absolute minimum, quantum tunneling – a phenomenon unique to particles on the quantum scale – allows this traveler to "tunnel" directly through hills to reach the absolute minimum, rather than have to metaphorically climb all the way back up the hill (Figure 2). Since aligning a set of data involves minimizing the distance between each pair of sequences – a textbook optimization problem – the MSA problem fits neatly into the functionality of quantum annealers [24].

1.3.2 Current Shortcomings. In spite of the potential advantage using quantum algorithms to find alignments may provide, there exist two major areas in need of immediate improvement. First, due to the relatively new nature of quantum annealers, existing algorithms tend to at best mirror rudimentary classical algorithms. That is, some algorithms mimic brute-force processes without the inclusion of more complex heuristics that aid the process, such as progressive or iterative techniques [24].

Secondly, modern quantum algorithms are constrained by hardware limitations [10]. The reliance on the quantum properties of particles leaves the qubits susceptible to slight changes in the environment [4]. These errors result in inconsistencies between the simulated solution and experimental results returned [21]. Furthermore, the number of quantum nodes available for public use is restricted, largely due to the limited size of existing computers. For example, the D-Wave quantum annealer Advantage, contains just over 5000 quantum bits [43] – barely meeting current supercomputing capabilities, and there exist few available annealers larger in size. This places an upper bound on the size of the test data. Thus issues are raised. The input datasets of genetic and protein sequences are large in both size and sequence length. So, a sufficient amount of qubit spin usage in these quantum computers is needed. The development of a more efficient tool for MSA capable of bypassing the constraints of hardware limitations is needed.

2 METHODS

We took inspiration from classical algorithms that utilize clustering methodologies [42], where sequences are grouped before being progressively processed via the alignment algorithm, providing a close approximation of the solution [11]. In short, we introduced classical-inspired processing methods to the quantum annealing process. To do so, we implemented an overarching progressive alignment structure throughout the algorithm.

We first determined the hardware on which the quantum algorithm could be run. This was used as a constraint to specify the algorithm body type. We then broke this project in three key parts: (1) Pre-processing, (2) Algorithm Body, and (3) Post-processing. These parts are defined by their function relative to the overarching algorithm, as outlined below and in Figure 3.

(1) The **Pre-processing [Key Modification]** part is the set of operations that reads in files and prepares the sequences for alignment.
- Read in sequences from FASTA file,
- Cluster sequences, and
- Convert sequence clusters into matrix.

(2) The **Algorithm body** returns the alignments of given set of sequences.
- Take in clusters and transform to form digestible by quantum solver and
- Align sequences per cluster.

(3) The **Post-processing [Additional Modifications]** processes the results obtained from Parts 1 and 2. in order to produce a final output for the user.
- Interpret the annealing results,
- Merge locally aligned clusters with previous alignments, and
- Output final alignment.

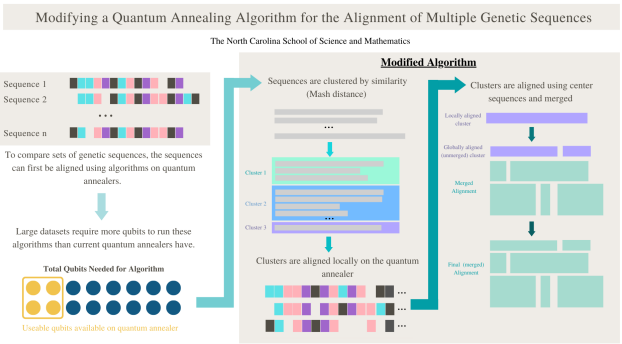


Figure 3: Visual obserview of MAQ algorithm approach (created by author).

2.1 Hardware

Thus, MAQ was run on the D-Wave Adiabatic Computing (Quantum Annealing) System, made accessible via the Leap integrated development environment (IDE). While other quantum annealers – including those developed by the New Energy and Industrial

Technology Development Organization [1], Ford Motor Cars, and Lockheed Martin [30] – exist, D-Wave annealers were selected due to their commercial availability and earlier establishment as a product available to the public [9].

Simulations of this system were also accessible. The Leap IDE is a quantum cloud service run using Python. The D-Wave Solvers may also be used locally. Here, D-Wave's Ocean v.5.2.0 software development kit [19] and dimod v.0.11.2 package [18] was used, allowing the quantum annealing environment to be simulated on the local system's central processing unit (CPU).

2.2 Pre-processing Development

2.2.1 Approaching Sequence Read-In And Storage. Prior to aligning the sequences, we parsed the sequences in from an external file. We assumed that the data – containing either protein or genetic sequences – is contained on a single FASTA Formatted Sequence file. Using the Biopython v1.79 package, the sequences were stored as Sequence Record objects, containing key information on the sequence's identity [5]. For the purposes of data storage and later processing, we assumed that every genetic sequence in the file had an unique identifier.

2.2.2 Introducing the Novel Modification. To implement the progressive alignment technique, we introduced a sequence clustering component to the pre-processing stage. To identify the clustering algorithm to accomplish this task, we first set the lowest possible bar. We observed the naive solution was not ideal. While the arbitrary assignment of sequences would not be costly, it would have come at the cost of the accuracy of the returned alignment. Thus a deliberate algorithm was sought for.

We took inspiration from the Feng-Doolittle progressive alignment approach [11]. The Feng-Doolittle algorithm uses classical computers to first group the clusters by similarity, then uses dynamic programming methods to merge the sequences following their local alignments [11], producing an approximate MSA. We used a simplistic approach, which was in line with a similarly inspired hierarchical clustering algorithm developed in 1988 [6].

More specifically, we adapted the ALFATClust algorithm and treated the clustering problem as a question of finding the nearest neighbor [3]. It used the Leiden algorithm to localize each cluster, connecting "communities" of these clusters based on relative similarity [39]. This differs from the greedy approach taken by most existing software tools, which are reliant on a limited set of parameters (thereby producing not ideal alignments).

To approximate the difference between sequences prior to clustering, ALFATClust uses the Mash (sample-based) technique [3]. While preliminary studies have shown the alternate, unsupervised learning-based algorithms, such as MeShClust, are able to process these sequences more rapidly [20], these algorithms return an unusually low number of clusters (with larger numbers of sequences per cluster) [3]. This is contrary to one of the primary objectives of MAQ, which seeks to reduce the total spin usage once these datasets are passed into the quantum annealers. The ALFATClust method holds its own against other algorithms that do not employ the Mash heuristic, demonstrating its viability for selection for our purposes [3].

Following initial testing, it was revealed that ALFATClust occasionally returns clusters that contain a small number of sequences (e.g. a 2-sequence dataset), for which calls on a quantum annealer may be deemed unnecessary. To remedy this, we introduced a minimum cluster threshold size. If a cluster size did not meet the threshold, it would be appended to the next cluster, the entirety of which was then aligned locally.

To create a standard of comparison across each subsequent sequence, we introduced a function to identify the centers of each of the clusters. The center was defined as a singular sequence in the group with the lowest total distance when compared against all other sequences in the cluster. This center re-emerges in the post-processing stage to aid in the merging of cluster alignments.

The Mash v.1.14 package was used to conduct preliminary estimations on the distances between each of the sequences [31]. The subsequent data was analyzed using the NumPy v.1.22.4 [14], SciPy v.1.8.1 [41], and Pandas v.1.4.2 packages [37]. The clustering algorithm calls on the Leiden algorithm v.0.8.10 package [39] and Python igraph v.0.9.11 package [7].

2.3 Main Algorithm

Each cluster is then passed through the main algorithm, with the center from a previously aligned sequence appended to the cluster for later merging. To implement the MSA problem in the annealing algorithm, we defined the problem space, developing the Hamiltonian for the distance minimization problem with constraints. The algorithms were thusly based on this problem formulation. While selecting the algorithm for the body, we considered three sets of variables: appropriate use of the (1) objective, (2) weights and penalties, and (3) constraints. Quantum spin usage was a secondary driving factor.

We defined (1) the objective to be the minimization the overall distance between the sequences. Thus, in constructing (2) the weights matrix, an effective method of comparison and storage must be used. Full penalties are applied in alignments where the elements in corresponding positions do not match. To avoid the insertion of unnecessary gaps, a partial penalty for these gaps are included. After the weights matrix in the sequences is found, (3) constraints may be applied. These constraints would be dependent on the approach.

We considered two potential approaches to problem formulation. We began by defining the parameters of the problem. When given a L-sized set of sequences with maximum sequence length N, the naive solution is to use a systematic brute-force approach. Every element in each sequence will compared against every other element in all other sequences. In this case, every possible pairing of elements will require a corresponding spin value to be stored. This requires a system on the magnitude of $O(N^L)$. This is by all means infeasible on current hardware, especially after gaps are inserted to account for element deletions or insertions (a biological phenomenon) [24].

After further research, we determined a secondary, more effective approach. Oscar Lindvall proposed using the Column Alignment Formulation (CAF) approach to align the sequences (Figure 4). It may be visualized using a table with L rows and some C columns. Given some user-defined parameter G, representing the maximum

Figure 4: CAF aligns sequences by assigning the elements in each sequence a column and inserting gaps into any column spaces with no elements [24].

number of gaps that may be inserted into the sequence to shift corresponding sections, C can be set equal to $N + G$. Every row represents a single sequence, and every column a single position. The goal, then, is to find the positions in every row where an element in the sequence can be placed, such that the number of differences per column is minimized. We assume G is significantly small relative to N, the maximum length of the sequence. In this case, the number of spins that must be represented at any point (that is, the number of qubits needed) is within $O(LN^2)$ [24], a reduction to manageable polynomial magnitude.

We used the CAF approach proposed by Lindvall, applying some pre-defined penalty g for the insertion of empty spaces in the sequences [24]. Per Lindvall's proposed algorithm, we constructed the matrix by comparing the sequences against one another each other, resulting in weights $w_{(s1,n1,s2,n2)}$ for every pairing of elements [24].

2.4 Post-Processing Development: Dynamic Programming

The output of the main algorithm is a matrix, where each row represents a sequence and each column a position [24]. The first '1' in the row is where the first element in the corresponding sequence is placed, the second '1' is where the second element is placed, and so on. If a '0' exists in the matrix, then a gap has been inserted in that position. We authored a simple method to interpret these results and transform them into readable strings.

However, the process at this step is incomplete. Alignments have only been made for the individual clusters. Recall that the alignment contains the center of the previous cluster. Using comparisons between the gaps inserted in the center in this and the previous iteration of the algorithm, we dynamically merge the clusters together, such that after merging, the current cluster is immediately forgotten from the quantum annealer.

3 RESULTS & COMPLEXITY ANALYSIS

To properly analyze the preliminary results returned by this modified algorithm, we reiterate that the main goals of this project were to

(1) Introduce classical-inspired heuristics to rudimentary quantum algorithms, and
(2) Reduce the spin usage per call of the quantum annealer.

In order to approximate the effectiveness of the algorithm in achieving this end, we conduct a rough space complexity analysis of the key impacts of (1) the weights determination function, (2) the quantum-dependent component, and (3) the merging function. These three areas have been impacted most strongly by the modifications.

3.1 Analysis of Weights Matrix Function

Let us consider an input dataset of L sequences, with a maximum sequence length of N and G inserted gaps per sequence. We first consider the characteristics of the initial algorithm for comparison. The creation of the weights matrix is especially consuming, since it requires storage of the comparisons between every individual element in the dataset. Since every possible pair of elements in distinct sequences is compared, the space complexity may roughly be given by

$$\begin{aligned} &O(\frac{N!}{2!(N-2)!} \times \frac{L!}{2!(L-2)!}) \\ &\simeq O(\frac{N(N-1)}{2} \times \frac{L(L-1)}{2}) \\ &\simeq O(N^2L^2) \end{aligned} \tag{1}$$

Let the clustering algorithm reduce the dataset to some number of clusters, such that the largest cluster has k sequences, where $k << L$. The weights matrix determination function is then reapplied to this reduced sample size, resulting in a complexity of

$$O(N^2k^2)$$

per cluster. However, the weights matrix must be applied at most k times. Therefore, the overall complexity of the weights matrix is given by

$$O(N^2k^2 \times \frac{L}{k}) \simeq O(N^2Lk). \tag{2}$$

Equation 2 presents a linear advantage over the initial weights matrix development requirements. However, this advantage is partially offset by the ALFATClust algorithm introduced during the pre-processing stage. Nevertheless, the ALFATClust's application of the Mash approximation for distance estimation cuts down significantly on the initial $O(N^2L^2)$ space complexity [3].

3.2 Analysis of Alignment Algorithm

Secondly, we consider the spin usage during the sequence alignment on clusters. Spin usage is a quantitative approximation of the number of nodes that will be used on the quantum annealer during computation. Recall we seek to reduce this usage per call of the quantum annealer.

The use of the Column Alignment Formulation (CAF) method already introduces a significant reduction on possible spin usage. The spin values – and resultant alignment – is stored in some matrix, where the number of columns is equal to the sum of the length of the sequence and number of gaps

$$C = N + G. \tag{3}$$

Thus, using Equation 3, we conclude the spin usage S is given by

$$S = C \sum_{i=1}^{L} N_i \tag{4}$$

where $N_1 ... N_L$ are the lengths of the sequences [24]. That is, the spin usage may be roughly described as

$$\begin{aligned} S &\in O(CLN) \\ &= O(LN(N+G) \\ &= O(L(N^2+NG) \\ &\simeq O(LN^2) \end{aligned} \tag{5}$$

where it is assumed $G << L$ [24].

We now consider the spin usage on a reduced number of sequences, given by k. Following a similar line of reasoning, the spin usage when run on a single cluster may be given by

$$O(kN^2) << O(LN^2) \tag{6}$$

Observe that the total spin usage (on the magnitude of $O(kN^2 \times \frac{L}{k}) \simeq O(LN^2)$ is not representative of the maximum spin usage at a single point, as the quantum annealer is called L/k distinct times. It is worth noting that this rough complexity analysis treats the processing time of the inputs as a black box, thereby not accounting for the space or time needed to translate the input system onto the corresponding architecture (that is, node arrangement) for the annealer. This approach is nonetheless effective, as it indirectly implies the net node usage on the quantum annealer. Therefore, it follows from this reasoning that an approximate linear advantage is achieved in spin usage.

3.3 Analysis of Merging Function

Lastly, we analyze the function that progressively merges the aligned clusters. The local alignments are stored in matrices containing the elements and gaps with their corresponding positions. These matrices have a maximum size $O(kC)$, meaning the space complexity for n clusters may be described as

$$\begin{aligned} &\sum_{i=0}^{n} k_i C \\ &\simeq O(LC) \\ &\simeq O(L(N+G)) \\ &\simeq O(LN+LG) \\ &\simeq O(LN) \end{aligned} \tag{7}$$

These clusters are aligned locally. The center of the previously aligned cluster is included in the alignment of the new cluster. When merging, this center serves as the metric of comparison and the entire sequence is iterated through at least once, with a maximum length of N, resulting in a minimum baseline runtime of $O(N)$. Additionally, any gaps (G) that are inserted are then propagated throughout the remainder of the corresponding alignment (including through the compiled sequences in all previous alignments). Thus, over n clusters, the total running time is approximately

$$\begin{aligned} &O(N) + \sum_{j=0}^{n} kGj \\ &\simeq O(N) + O(kGn(n+1) \\ &\simeq O(N) + O(kGn^2) \\ &\simeq O(N + kGn^2) \end{aligned} \tag{8}$$

In the worst case scenario, to draw an upper bound on the runtime, $n = L$ and $k = 1$. Then, the worst case runtime is roughly

$$O(N + GL^2) \tag{9}$$

4 TESTING THE ALGORITHM

The developed algorithm, named MAQ, was run on a small, sample dataset for comparison (Table 2). Throughout the development process, the algorithm was repeatedly tested on this reduced dataset. Each sequence in the dataset was a derivation of some "base" sequence that represented some accepted sequence, along with an identical sequence "control" that ensured the most basic alignment (of the same sequences) could be achieved. Each subsequent sequence then contained at least one fundamental mutation that may occur in generic sequences (e.g. insertion, deletion, or point mutations). The sequences are identified in Table 2 accordingly. When run on ALFATClust, the dataset is clustered into three distinct sets of sequences, making it ideal to test the clustering-based MAQ algorithm.

Table 2: Alignment returned by MAQ algorithm using sample dataset (created by author).

ID	Sequence Alignment								
Base	-	N	V	R	L	M	L	R	L
Control	-	N	V	R	L	M	L	R	L
Insertion	M	N	V	R	L	M	L	R	L
Deletion	-	N	-	R	L	M	L	R	L
Point	-	N	V	M	L	R	L	N	L
InsertionAndDeletion	M	N	V	R	L	-	R	-	L

Table 3: Alignment returned by Oscar Lindvall's algorithm [24] using sample dataset (created by author).

ID	Sequence Alignment								
Base	N	V	-	R	L	M	L	R	L
Control	N	-	V	R	L	M	L	R	L
Insertion	M	N	V	R	L	M	L	R	L
Deletion	N	-	R	L	-	M	L	R	L
Point	N	-	V	M	L	R	L	N	L
InsertionAndDeletion	M	N	V	-	R	-	L	R	L

Table 4: Alignment returned by Kalign [23] using sample dataset (created by author).

ID	Sequence Alignment								
Base	-	N	V	R	L	M	L	R	L
Control	-	N	V	R	L	M	L	R	L
Insertion	M	N	V	R	L	M	L	R	L
Deletion	-	-	N	R	L	M	L	R	L
Point	-	N	V	M	L	R	L	N	L
InsertionAndDeletion	M	N	V	R	L	R	L	-	-

4.1 Metrics of Comparison

We firstly define the metrics used to compare these three MSA tools. We quantify the effectiveness of the algorithm by considering the alignment's deviation from the norm. The analysis is considered by column (following the CAF methodology), with pairwise comparisons conducted. In other words, we use a sum-of-pairs scoring method. For every pair of elements that differ in the same column, the total score for the alignment is incremented by +1, although differences between base pairs or amino acids and gaps will have no penalty (an adjustable parameter during the development of the problem space). An ideal alignment will have a total score of 0. The greater the alignment score, the less effective the alignment.

We now define this alignment score formally. Let us label the sequences in the final alignment from $\{s_0, s_1, ..., s_L\}$, organized in a matrix containing C columns and L sequences. Note these aligned sequences include any gaps inserted after the dataset is processed using the alignment algorithm. Then, construct a new matrix A with dimensions $C \times L \times L$, where the element $a_{c,i,j} \in A$ equals 1 if the cth element of sequences s_i and s_j are not equivalent and are not gaps and 1 otherwise. Then, the alignment score is defined as

$$\sum_{c=1}^{C} \sum_{i=1}^{L} \sum_{j=i}^{L} A_{c,i,j}. \tag{10}$$

For example, consider the set of sequences AT, T. An example alignment may be seen in Table 5. Observe that the first column has 3 pairs of alignments that do not match. The pair (A, T) has weight +1, while the pairs $(A, -), (-, T)$ do not match but contain gaps, so these differences are weighted at 0. Observe that the second column does not contain any pairwise differences. Using Equation 10, we find the score of the alignment in Table 5 is 1.

Table 5: Sample genetic sequence alignment, with a resultant alignment score of 1 (created by author).

A	T
-	T
T	T

4.2 Comparing with Existing Algorithms

We conducted preliminary tests on MAQ and compared the results obtained against results from two other algorithms: the unmodified Lindvall algorithm and a classical algorithm that uses similar progressive techniques. Much like how MAQ clusters sequences into local groups prior to alignment, Kalign focuses on alignments in local regions [25], employing a heuristic version of the Wu-Manber string (sequence) alignment algorithm. Kalign was shown to be significantly more accurate than other methods on large datasets, especially when compared against popular methods, such as Balibase and Prefab [23]. The algorithm was an estimated 10 times faster than ClustalW, an algorithm that makes use of tree-like data structures (arguably a more sophisticated form of clustering) to align the sequences [38].

After the test dataset of sequences (as seen in Table 2) was aligned on the three algorithms (MAQ, Lindvall's, and Kalign), the alignment scores were calculated using Equation 10 and organized in Table 6.

Table 6: MAQ is able to return an alignment with competitive alignment scores on relatively small sets of sequences (created by author).

Algorithm	Alignment Score by Column									Total
	1	2	3	4	5	6	7	8	9	
MAQ	0	0	0	5	0	4	5	4	0	18
Lindvall	8	2	4	7	4	4	0	5	0	34
Kalign	0	0	5	5	0	8	0	4	0	22

5 MAJOR CONCLUSIONS

The world of bioinformatics shapes societal responses to disease. A significant part of this understanding arises from pattern identification, which may be used to find information to predict how patients may respond to various diseases or treatments. This poses a series of sequence-based problems that are solvable on algorithms. Among these, MSA plays a significant role. After all, comparison of large sets of genetic or protein sequences is reliant on the assurance that these sets have been aligned in a logical manner. In spite of its relevance, the problem is NP-complete, which speaks to the need for the development of algorithms that are capable of stepping beyond the 0's and 1's of today's classical computers. Our developed algorithm, MAQ, is one step in such this direction.

The application of quantum computing to problems is not new [27]. Over the years, algorithms for tasks such as genetic sequencing and protein structure prediction have been proposed [27]. However, many are heavily restricted by spin usage and the relatively new nature of the field. MAQ introduces a classical-inspired approach reduces the spin usage per call of the quantum annealer.

The algorithm first clusters the sequences using ALFATClust [3]. The reduced sequence sets are then compared and aligned on the main algorithm, modified from Oscar Lindvall's approach [24]. The resultant alignments are then dynamically merged based on the relative spacing of the center sequences of each cluster. The final, progressively aligned alignment is then returned to the user.

A linear advantage of $O(L/k)$, given L total sequences and k clusters, is achieved in the reduction of spin usage per call of the quantum computer (Equation 2). However, added complexity due to the addition of the clustering step and repetitive calls to the quantum annealer adds to the overarching running time. Nonetheless, the spin usage of each single call on quantum annealers has been reduced. This allows for the adjustment of large datasets for current quantum hardware that has yet to be able to handle significant space usage without significant loss of information.

Furthermore, when run on a test dataset, MAQ was shown to be comparable to existing MSA algorithms, including Lindvall's initial algorithm and Kalign. For this specific dataset, MAQ performed better, with an alignment score of 18, relative to the scores of 34 and 22 for Lindvall's algorithm and Kalign, respectively. Thus, it is comparable to existing algorithms.

6 DISCUSSION AND WIDER APPLICATIONS

One must caution that the advantage achieved by MAQ is dependent on the characteristics of the data. The viability of a clustering-based method may be determined using the rank of the set of sequences (that is, how similar the sequences are to each other, where lower rank suggests larger similarity). The lower the rank of the set, the more likely the results will resemble that of Lindvall's algorithm, since the number of clusters is reduced. MAQ assumes there exists sufficient distinctions between each sequence in the set such that they may be clustered into a reasonable number of subsets. In other words, there is moderate variability between the sequences. In the case all of the sequences are nearly identical (say, with an estimated similarity of > 0.99, the clustering may be deemed ineffective. Consider the alternative extreme. In the case the sequences have unusually high rank (where the variability between the sequences is high), the number of clusters will be close to the initial number of sequences, and the impact of the clustering algorithm will be called into question. One may argue the sequences in these extreme cases should instead be grouped by the order it is read in from the file. This would consume fewer resources.

Future research is needed to quantify the actual effectiveness of clustering prior to alignment. This is especially important since the clustering algorithm is costly, as it is itself tackling a NP-hard problem [26]. further study may reveal a definitive response on whether the cost of clustering the sequences exceeds the benefits of an improvement in alignment when compared against, for instance, alignment of random groupings of sequences.

Furthermore, future MAQ versions may explore other algorithms, including those that use K-means clustering, where the number of clusters is predefined. Then, the approximate reduction of the spin usage per call on the quantum annealers may be approximated with greater certainty. Granted, although the number of cluster will be guaranteed (including for sets with low rank), the size of these clusters will still be dependent on user parameters.

This algorithm deserves further revisitation. Tackling MAQ as three distinct components that funnel into one another presents an opportunity for improvement. Additional research is needed to investigate approaches to consolidating sequence clustering and alignment, especially with regards to the creation of the weights matrix (a costly process). For example, the pairwise distance of sequences is first estimated using the Mash heuristic during the clustering pre-processing phase. The pairwise comparisons are then completed a second time while creating the problem space the quantum annealer will solve (although the exact mechanics differ). Thus, a standalone clustering algorithm may not be the best integration into MAQ. Rather, future versions of MAQ may look to consolidate these pairwise comparisons to reduce overall iterations through the sequences. Alternative approaches should also be studied.

Additionally, when run on small datasets, the quantum annealing-based algorithms may regularly return different results. This is likely the result of multiple "lowest energy state" configurations. In MAQ, these differences may be propagated across the clusters, magnifying minor decisions early on in the alignment process between mathematically-identical alignment states. For each cluster aligned, there is no guarantee that the arbitrarily chosen state will result in the lowest alignment score across the total alignment. It merely guarantees a low alignment score locally. In order words, the dynamic merging process assumes all previous alignments are ideal, an assumption that does not always hold true. Despite this, this characteristic of the algorithm may be harnessed as an advantage. For example, rather than returning a single plausible alignment, several alignments – one corresponding to each combination of the ideal, local solutions – may instead be simultaneously compared by the algorithm. This may open the door for a more accurate final solution to be returned. Further research is needed to explore alignment algorithms that may make the most of the existence of a set of plausible local alignment results.

MAQ demonstrates the viability of quantum computing as a supporting system for studies into computational biology. This is a part of the wider driving force that dictates the possible paths of research development. After all, MSA is just one of many optimization problems in bioinformatics. Genetic engineering and sequencing, for example, are heavily reliant on the capabilities of existing technology. These capabilities are defined by the accuracy and accessibility of these tools. As the accessibility of quantum computers increases, a rising number of algorithms – including MAQ – are bridging the gap between quantum computing and other areas.

These identified areas have the potential to impact millions of human lives. Chief among them are epidemiological and phenological studies. In particular, comparison of these sequences permits for a stronger understanding of the human genome. More rapid sequencing tools will help translate compiled genomic data into medically useful information [13]. This includes a better approach to treatment response prediction – including chemotherapy – and phenology determination of disease strains. Through extensive multiple sequence analysis (made possible through alignment), medical professionals' understanding of the genetic patterns corresponding to phenotypical characteristics may be expanded. These developments have the potential to impact the 33.4 million individuals who pass through the US hospital system annually [2], along with the countless others who use any form of healthcare service. In order to achieve this, however, refinement of the quantum annealing process and algorithms must be conducted. As problem sizes continue to grow and the need for algorithms with lower space complexity and runtimes continues, heuristics such as that taken by MAQ will continue to emerge, marking this as an area of strong potential, worthy of further research.

7 STUDENT REFLECTION

MAQ was the result of a 9-month student research project I (the author) conducted. The investigative project explored the plausibility of applying quantum computing as a tool. In particular, I focused on addressing current limitations of the quantum hardware. However, arriving at this focus involved a rather indirect path consisting of a series of decisions.

My initial research had led me into a more abstract form of string alignment. This pure mathematics problem approached the situation via graph theory and employed techniques beyond the scope of this paper. I had initially begun with the intention of applying my previous understanding of quantum-inspired and quantum

computing algorithms to the project. Yet these purely theoretical subjects felt disconnected from ongoing problems in the world, and I struggled to identify a path forwards.

Over time, as the research plan began to solidify, I encountered an increasing number of applications for these algorithms. The puzzle pieces began to fall into place as I read about the application of MSA algorithms to genetic sequencing. I found I was revisiting a subject that had fascinated me years prior, and my appreciation for interdisciplinary studies grew.

This played a role in reshaping my long-term plans for study. In particular, my focus transitioned from pure mathematics and theoretical computer science to computational biology. While the two former fields are still on my radar as fields of interest, I recognize computational biology will likely play a larger role in the direction I take for future endeavors. Following conversations with a number of current graduate students, professors, and researchers in the field, I hope to go into and remain in research and academia following my college and (ideally) graduate studies.

Nevertheless, I recognize this research project is merely a small glimpse of what is plausible in the realms of bioinformatics and quantum computing. Even as I have gained a stronger understanding of algorithmic thinking, implementing quantum annealing, and the mathematics surrounding the fields, I realize I have much more left to learn. I have no intention of stopping my curiosity, and I hope to continue to expand my understanding of what is possible over the next few decades.

ACKNOWLEDGEMENTS

Thank you to Mr. Robert Gotwals for providing feedback and guidance throughout the research process.

REFERENCES

[1] 2018. Members selected for the Research Program for the development of a quantum annealing machine enabling high-efficiency, high-speed processing. https://www.nec.com/en/press/201812/global_20181212_03.html

[2] 2022. Number of individuals who pass through the US hospital system annually.

[3] Jimmy Ka Chiu and Rick Twee-Hee Ong. 2022. Clustering biological sequences with dynamic sequence similarity threshold. *BMC Bioinformatics* 23, 1 (2022). https://doi.org/10.1186/s12859-022-04643-9

[4] I. L. Chuang, R. Laflamme, P. W. Shor, and W. H. Zurek. 1995. Quantum computers, factoring, and decoherence. *Science* 270, 5242 (1995), 1633–1635. https://doi.org/10.1126/science.270.5242.1633

[5] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25, 11 (2009), 1422–1423.

[6] F. Corpet. 1988. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research* 16, 22 (1988), 10881–10890. https://doi.org/10.1093/nar/16.22.10881

[7] Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal* Complex Systems (2006), 1695. https://igraph.org

[8] D-Wave. [n. d.]. What is quantum annealing? https://docs.dwavesys.com/docs/latest/c_gs_2.html

[9] Cem Dilmegani. 2019. Quantum Annealing in 2022: Practical quantum computing. https://research.aimultiple.com/quantum-annealing/

[10] Marco Fellous-Asiani, Jing Hao Chai, Robert S. Whitney, Alexia Auffeves, and Hui Khoon Ng. 2021. Limitations in quantum computing from resource constraints. *PRX Quantum* 2, 4 (2021). https://doi.org/10.1103/prxquantum.2.040335

[11] Da-Fei Feng and Russell F. Doolittle. 1987. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of Molecular Evolution* 25, 4 (1987), 351–360. https://doi.org/10.1007/bf02603120

[12] Konstantinos Giannakis, Christos Papalitsas, Georgia Theocharopoulou, Sofia Fanarioti, and Theodore Andronikos. 2019. A Quantum-inspired optimization Heuristic for the Multiple Sequence Alignment Problem in Bio-computing. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 1–8. https://doi.org/10.1109/IISA.2019.8900740

[13] Claudia Gonzaga-Jauregui, James R. Lupski, and Richard A. Gibbs. 2012. Human genome sequencing in health and disease. *Annual Review of Medicine* 63, 1 (2012), 35–61. https://doi.org/10.1146/annurev-med-051010-162644

[14] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[15] Halsted R. Holman. 2020. The relation of the chronic disease epidemic to the Health Care Crisis. *ACR Open Rheumatology* 2, 3 (2020), 167–173. https://doi.org/10.1002/acr2.11114

[16] Sandeep Hosangadi. 2012. Distance Measures for Sequences. *arXiv* (Aug 2012). https://doi.org/10.48550/arXiv.1208.5713

[17] Ming Huang, Nilay D. Shah, and Lixia Yao. 2019. Evaluating global and local sequence alignment methods for comparing patient medical records. *BMC Medical Informatics and Decision Making* 19, S6 (2019). https://doi.org/10.1186/s12911-019-0965-y

[18] "D-Wave Systems Inc.". 2022 [Online]. *D-Wave Systems Dimod Package*. https://github.com/dwavesystems/dimod

[19] "D-Wave Systems Inc.". 2022 [Online]. *D-Wave Systems Ocean SDK*. https://github.com/dwavesystems/dwave-ocean-sdk

[20] Benjamin T James, Brian B Luczak, and Hani Z Girgis. 2018. Meshclust: An intelligent tool for clustering DNA sequences. *Nucleic Acids Research* 46, 14 (2018). https://doi.org/10.1093/nar/gky315

[21] Scott Johnstun and Jean-François Van Huele. 2021. Understanding and compensating for noise on IBM Quantum Computers. *American Journal of Physics* 89, 10 (2021), 935–942. https://doi.org/10.1119/10.0006204

[22] Nikolai Klebanov. 2018. Genetic predisposition to infectious disease. *Cureus* (2018). https://doi.org/10.7759/cureus.3210

[23] Timo Lassmann and Erik LL Sonnhammer. 2005. Kalign – an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics* 6, 1 (2005). https://doi.org/10.1186/1471-2105-6-298

[24] Oscar Bulancea Lindvall. 2019. Quantum Methods for Sequence Alignment and Metagenomics. *KTH Royal Institute of Technology* (2019). https://doi.org/smash/get/diva2:1345195/FULLTEXT02

[25] Fábio Madeira, Matt Pearce, Adrian R Tivey, Prasad Basutkar, Joon Lee, Ossama Edbali, Nandana Madhusoodanan, Anton Kolesnikov, and Rodrigo Lopez. 2022. Search and sequence analysis tools services from EMBL-EBI in 2022. *Nucleic Acids Research* 50, W1 (2022). https://doi.org/10.1093/nar/gkac240

[26] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. 2012. The planar K-means problem is NP-hard. *Theoretical Computer Science* 442 (2012), 13–21. https://doi.org/10.1016/j.tcs.2010.05.034

[27] Vivien Marx. 2021. Biology begins to tangle with quantum computing. *Nature Methods* 18, 7 (2021), 715–719. https://doi.org/10.1038/s41592-021-01199-z

[28] FN Muhamad, RB Ahmad, SM Asi, and MN Murad. 2018. Performance analysis of Needleman-Wunsch algorithm (global) and Smith-Waterman algorithm (local) in reducing search space and time for DNA sequence alignment. *Journal of Physics: Conference Series* 1019 (2018), 012085. https://doi.org/10.1088/1742-6596/1019/1/012085

[29] Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48, 3 (1970), 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

[30] Ann Obata. 2022. Companies building and exploring applications with quantum annealing. https://quantumzeitgeist.com/companies-building-and-exploring-applications-with-quantum-annealing/

[31] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. 2016. MASH: Fast genome and metagenome distance estimation using MinHash. *Genome Biology* 17, 1 (2016). https://doi.org/10.1186/s13059-016-0997-x

[32] Kathryn A. Phillips, Julia R. Trosman, Robin K. Kelley, Mark J. Pletcher, Michael P. Douglas, and Christine B. Weldon. 2014. Genomic sequencing: Assessing the health care system, policy, and big-data implications. *Health Affairs* 33, 7 (2014), 1246–1253. https://doi.org/10.1377/hlthaff.2014.0020

[33] Max Roser and Hannah Ritchie. 2021. Burden of Disease. *Our World in Data* (2021). https://ourworldindata.org/burden-of-disease.

[34] Jeong Seop Sim and Kunsoo Park. 2003. The consensus string problem for a metric is NP-complete. *Journal of Discrete Algorithms* 1, 1 (2003), 111–117. https://doi.org/10.1016/s1570-8667(03)00011-x

[35] Andrew Steane. 1997. Quantum Computing. *arXiv* (Aug 1997). https://doi.org/10.48550/arXiv.quant-ph/9708022

[36] Ewin Tang. 2019. A quantum-inspired classical algorithm for recommendation systems. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019). https://doi.org/10.1145/3313276.3316310

[37] The Pandas Development Team. 2020. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134

[38] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. 1994. Clustal W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22, 22 (1994), 4673–4680. https://doi.org/10.1093/nar/22.22.4673

[39] V. A. Traag, L. Waltman, and N. J. van Eck. 2019. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports* 9, 1 (2019). https://doi.org/10.1038/s41598-019-41695-z

[40] D. Ventura and T. Martinez. 1998. Quantum associative memory with exponential capacity. *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)* (1998). https://doi.org/10.1109/ijcnn.1998.682319

[41] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[42] Yingying Wang, Hongyan Wu, and Yunpeng Cai. 2018. A benchmark study of sequence alignment methods for protein clustering. *BMC Bioinformatics* 19, S19 (2018). https://doi.org/10.1186/s12859-018-2524-4

[43] Dennis Willsch, Madita Willsch, Carlos D. Gonzalez Calaza, Fengping Jin, Hans De Raedt, Marika Svensson, and Kristel Michielsen. 2022. Benchmarking advantage and D-wave 2000Q quantum annealers with exact cover problems. *Quantum Information Processing* 21, 4 (2022). https://doi.org/10.1007/s11128-022-03476-y

[44] Shenggen Zheng, Daowen Qiu, and Jozef Gruska. 2017. Time-space complexity advantages for quantum computing. *Theory and Practice of Natural Computing* (2017), 305–317. https://doi.org/10.1007/978-3-319-71069-3_24

An Educational and Training Perspective on Integrating Hybrid Technologies with HPC Systems for Solving Real-World Commercial Problems

Stefano Mensa
The Hartree Centre, STFC
Warrington, Cheshire
stefano.mensa@stfc.ac.uk

Emre Sahin
The Hartree Centre, STFC
Warrington, Cheshire
emre.sahin@stfc.ac.uk

George Williamson
The Hartree Centre, STFC
Warrington, Cheshire
george.williamson@stfc.ac.uk

Robert J. Allan
The Hartree Centre, STFC
Warrington, Cheshire
robert.allan@stfc.ac.uk

ABSTRACT

Delivering training and education on hybrid technologies (including AI, ML, GPU, Data and Visual Analytics including VR and Quantum Computing) integrated with HPC resources is key to enable individuals and businesses to take full advantage of digital technologies, hence enhancing processes within organisations and providing the enabling skills to thrive in a digital economy. Supercomputing centres focused on solving industry-led problems face the challenge of having a pool of users with little experience in executing simulations on large-scale facilities, as well as limited knowledge of advanced computational techniques and integrated technologies. We aim not only at educating them in using the facilities available, but to raise awareness of methods which have the potential to increase their productivity. In this paper, we provide our perspective on how to efficiently train industry users, and how to engage with them about wider digital technologies and how these, used efficiently together, can benefit their business.

KEYWORDS

Education, Training, HPC, Integrated Technologies, Customer Success, Quantum Computing Training, GPU Training, Digital Twinning

1 INTRODUCTION

The Hartree Centre (HC) is part of the Science and Technology Facilities Council (STFC) - one of UK Research and Innovation's research councils - building on the rich established scientific heritage and a network of international expertise to support the UK's continued leadership in computational science and digital innovation [1, 4, 6]. HC supports businesses and organisations of any size in the UK in exploring and implementing technologies such as supercomputing (HPC), data analytics and artificial intelligence (AI) for increased productivity, cleverer innovation, and economic growth. The centre is home to some of the most cutting-edge digital technologies and experts in the UK, supported by sizeable UK government funding and strategic partnerships with industry leaders. In 2021, the Hartree National Centre for Digital Innovation (HNCDI) [2] programme was established to provide a safe and supportive environment for UK businesses and public sector organisations to acquire the skills needed to adopt AI, develop proofs-of-concept and de-risk investment into emerging digital technologies such as quantum computing.

The Hartree Centre's in-house skills set is key to helping industrial partners deliver solutions to real-world challenges. True customer success, however, is achieved when customers fully understand how to apply acquired knowledge and can adapt it to their own business needs. Dealing with industrial customers as end users of HPC facilities can present unique challenges. A typical user coming from an industrial background is remarkably knowledgeable in a specific domain area, however, they often lack the knowledge required to perform numerical simulations on large-scale computing facilities. Furthermore, the adoption of hybrid computational technologies (that is the use of computational techniques such as ML, AI etc. in combination with classical HPC) is hindered by the lack of detailed understanding of the functionality.

These two issues have often three negative outcomes. The first casualty due to lack of "operational knowledge", is productivity. Users that do not know their way in an HPC infrastructure usually end-up in using the facility in a sub-optimal way, hence resulting in loss of productivity and ultimately financially impacting the project itself. Indeed, poor usage of computational resources will unavoidably drain paid project compute time allocation. Second, inexperienced, and non-self-sufficient users impact data-centres operations, opening tickets and incidents that take time to solve, diverting staff time into non-critical troubleshooting. Finally, lack of understanding of the low level functionality of new technologies limits their uptake and hinders digital innovation in the business. Hartree Centre staff aim to assist in all three areas.

In this paper, we provide our perspective on how to efficiently train users with an industrial background, not only on how to use HPC systems but also on how to engage with them about wider digital technologies and how these, used efficiently together, can

https://doi.org/10.22369/issn.2153-4136/14/1/6

benefit their business. Here, we describe our training and education strategy for users with a core industrial background using the following three stages. The first stage is about building a confident and self-sufficient cohort by providing consistent and systematic training on how to use HC supercomputing facilities. These users can pass on knowledge to their colleagues. The second stage is designed to build digital innovation awareness, where we engage with customers by showcasing successful examples of integration of hybrid technologies within a business pipeline, especially by means of visualisations aids. The final stage is a more specialised training and education, where customers already aware of the benefits of digital innovation for their business can gain in-depth knowledge via the HNCDI Explain programme.

1.1 HNCDI: The Hartree National Centre for Digital Innovation

The Hartree National Centre for Digital Innovation is a collaborative programme with IBM which will enable businesses to acquire the skills, knowledge and technical capability required to adopt digital technologies like supercomputing, data analytics, artificial intelligence (AI) and quantum computing.

Through HNCDI we provide a safe and supportive environment for organisations to explore the latest digital technologies and skills, develop proofs-of-concept and apply them to industry and public sector challenges. Our dynamic and collaborative approach is driven by industry requirements and will help organisations to de-risk investment in new and emerging digital technologies.

Either at the start of their digital journey or trying to advance to the next level, we can help businesses navigate the possibilities of AI and quantum computing technologies to discover the next step for their digital development.

The HNCDI programme is divided in four work-streams.

(1) *Emerging Technology:* we are looking at the future of computing in the UK, helping businesses to identify the areas where emerging digital technologies like quantum computing might offer the most competitive advantage.
(2) *Excelerate:* through our applied industrial research, we help to turn good ideas into industry-ready solutions that address business challenges, embedding AI solutions across the industry.
(3) *Explore:* it aims to go one step further by finding ways to solve industry challenges when there isn't an existing off-the-shelf solution but there is evidence it can be solved and a business value and motivation to solve it.
(4) *Explain:* in this work-stream, HC staff works with individuals to identify learning pathways through our course catalogue that will equip their organisation with the skills needed to take advantage of digital technologies. Explain will be discussed more in-depth in the following Sections.

2 BUILDING A CONFIDENT AND SELF-SUFFICIENT USER COHORT

Successful routine use of a supercomputer in a commercial project goes hand in hand with the proficiency of its project members in making the most of the available infrastructure. This involves managing their data as well as efficiently targeting the resources in terms of processor type, number, etc. for specific simulation cases. Although HPC infrastructure across the world operates with the same principles (distribute computing over a fast network, hybrid hardware, job scheduler to orchestrate the workload, distributed file system etc.) and use pretty much the same family of operating systems (e.g. Linux based clusters), each data centre is different and only rarely will HPC systems have identical features. Thus, even experienced users will have somewhat to re-learn and adapt when moving onto a new supercomputer. The time taken to adapt to a new machine depends on the proficiency of the user. There are several educative and training perspectives as mentioned in [11, 14, 26–29].

We believe that customer on-boarding plays a crucial role for businesses' journey towards integration of hybrid technologies. For this reason, each and every new HC user undergoes an on-boarding process that we call "driving license", a training course delivered as a two hour lecture in which the users are expected to learn the fundamentals of our supercomputer, Scafell Pike (Top 500 list). To complement the lecture, a hand-book is also made available, see [19]. Topics covered in the course span both hardware and software of the machine, and the message we want to share is that there is no efficient usage of a supercomputer if first it is not clearly understood how the machine works, in terms of it's hardware, the nature of the file-system, the job-scheduler and the overall software stack. Finally, practical examples of job submissions are provided, also useful as a starting template for customised submission scripts. Before users receive their machine accounts, a "driving license" test needs to be passed, in which the users demonstrate competency in the usage of the machine, by being quizzed on a number of question regarding our HPC facility usage.

The HC on-boarding process has a number of benefits. First, the training provided guarantees a consistent minimum working knowledge across users in the centre, meaning that even a complete novice possess the relevant knowledge to comfortably move around the system. Second, the amount of downtime that users experience due to unexpected issues at submission time, execution time and so forth is significantly reduced, as understanding of the system provides the user with basic diagnostic skills (e.g. the job did not produce any output because it was submitted from the wrong base directory). Third, data-centre operations also benefits a trained user workforce a, generally speaking, trained users tend to raise less tickets for issues, thus reducing the amount of staff time spent troubleshooting basic issues.

3 BUILDING DIGITAL INNOVATION AWARENESS

The second fundamental stage of training and educating users with an industrial background is to build digital innovation awareness, that is to understand which and how digital technologies can enhance business productivity and profits. However, learning about the power of computational methods and, in general, novel technologies to address business critical objectives is hard if approached under a purely theoretical perspective, and a practical, tangible example would be a more effective learning tool. The HC has 10 years of experience in enhancing businesses profitability through the application of advanced technologies, with a large

portfolio of projects and case studies available to share. Thanks to our portfolio, we built a number of demonstrations showcasing projects outcomes obtained with hybrid technologies (HPC, AI/ML etc.), using visual computing strategies aiming at immersing the customer in an engaging visual and virtual environment, to help promoting and understanding the impact of a knowledgeable use of such hybrid technologies. We found out that visual computing is a good enabler towards digital innovation awareness.

3.1 Visual Computing as an Educational Tool

Visual computing and digital twinning have unquestionable advantage to bring insight and deeper understanding for scientific, industrial and educational fields. Exploiting such emerging technologies can allow us to bring more awareness and interest to applications of HPC.

Here at the Hartree Centre we have a range of visual computing facilities suitable for demonstration and training purposes as well as use in a project work. Our two showcase rooms house large main displays, and both of these displays are stereo 3D capable and both are equipped with high-end visualisation workstations and HD audio systems. All the displays and devices throughout the Visual Computing suit are connected through a high-speed multi-cast network utilising Crestron NVX transducers and switches such that any computer in the system can be connected to any display. We typically connect to our HPC facilities through VNC [5] for interactive access and remote visualisation. We have a number of licensed and open source software tools available for our scientists, engineers and partners to train and utilise for project work. Combining tools such as these with our super-computing systems can allow us to manipulate, control and visualise data on a massive scale for educational and training purposes.

Ranging from partners to students, we demonstrate a number of visually exciting case studies from past projects to help raise awareness of how we can engage with them and help them realise their potential. Our visualisation systems are instrumental in improving the quality of demonstrations and presentations, and this helps us to enable better experience during training and education processes.

One of the projects we demonstrate in this way is our Virtual Wind Tunnel project (VWT) [30]. We visualised a project investigating airflow over a prototype car body, and it produced some near photo-realistic renders of streamlined data from computational fluid dynamics (CFD) simulations. We ran the CFD simulation on one of our supercomputer systems and the data produced was post-processed and overlaid onto the CAD model in the rendering software also running on the supercomputer. This led to the development of a supercomputer CFD workflow, which we call our Virtual Wind Tunnel (VWT) and an application used to display and investigate CFD data overlaid on CAD models in a more realistic and human-relatable 3D/VR environment. The app simulates an actual wind tunnel and allows the user to move around the test object and view it from all angles while displaying streamlined data. Particles can be added to the streams and animated along the streamline trajectories to show the development of flow. This work-in-progress app when shown on our larger displays gives a more realistic and relatable feel when visualising data. This in turn demonstrates to our customers how the correct visualisation can give insight into what the data is showing us by putting the human back in the loop.

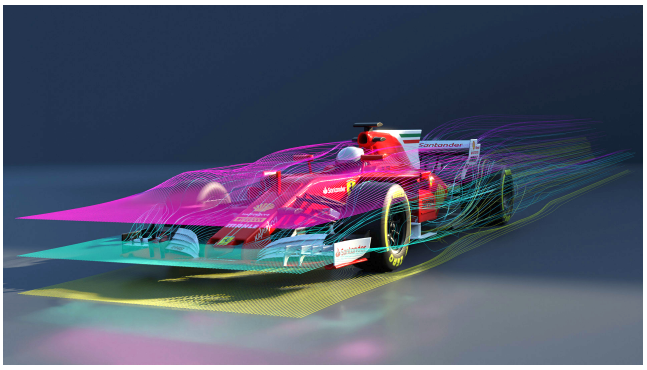


Figure 1: Photorealistic rendering of a F1 Ferrari overlapped with CFD streamline data simulated on our HPC facility.

Another example of visualising big data is our Dengue Virus demonstration. Here we show a stereo 3D visualisation of the viral protein that causes dengue fever on our 4.7-metre display pointing out to people that this is just one way to view the 1.1 million atoms that make up the protein. Such kind of protein visualisation is helpful to intuitively grasp the structure of the virus, potentially identifying hidden pockets that could be targeted by drugs.

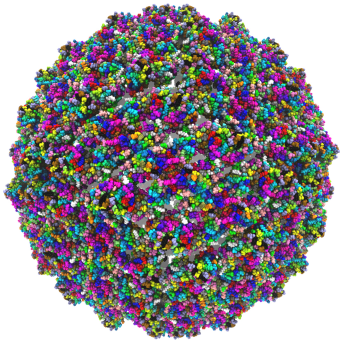


Figure 2: Dengue virus envelope protein rendered with VDW drawing method using VMD [16].

Demonstrating techniques for visualising big data sets in this way will hopefully show a good visualisation can bring potential benefits to educational, scientific and industrial projects and how it can help to make big data more accessible to more people.

4 SPECIALIST TRAINING AND EDUCATION WITH THE HNCDI EXPLAIN PROGRAMME

To meet some of the above goals, we offer application-focused training via the HNCDI Explain program, designed to enable individuals and businesses to take full advantage of digital technologies to enhance processes within their organisation and provide the skills that enable them to thrive in a digital economy.

Our training model is flexible and built with industry in mind. Whether learning the basics or searching for new tools and techniques to apply, a combination of self-directed online learning and face-to-face practical sessions can be used with certification.

The HC team will work with individuals to identify learning pathways through our course catalogue that will equip their organisation with the skills needed to take advantage of digital technologies. These skills can then be passed on to colleagues who will also have free access to the training materials. In general, four different levels of training are provided:

- *Introductory:* trainees from a non-related background with very little knowledge of the subject area;
- *Learner:* trainees with some theoretical or practical knowledge within the relevant subject area;
- *Independent user:* trainees who can work independently within the subject area but would require guidance for solving complex problems;
- *Practitioner:* trainees actively working in the subject field, looking to investigate emerging technology developments, and new techniques and/or develop collaborative multidisciplinary applications with higher levels of complexity.

Below two examples of specialist training we provide. A more comprehensive list of training courses offered can be found here [3].

4.1 Computational use of GPUs

Much research software, particularly open source, is nowadays developed to work using GPUs for the acceleration of critical numerical components. This is particularly true in fields such as machine learning and AI, bio-informatics and chemistry, solution of linear systems of equations for engineering applications and so forth. Despite the technological relevance, good software engineering practice for GPU accelerated software is somewhat limited to a small portion of specialised software engineers, and it's only being taught in specific academic degree courses, leaving interested users to rely on online open-source material or resorting to a self-taught strategy.

HC has a core specialism in GPU software development, as well as owning large GPU-based resources. Such specialism and hardware are exploited by the Centre to train users, aiming at sharing good GPUs software engineering practices. Teaming with partners such as NVIDIA, we offer hands-on training in GPU accelerated computing to solve real-world and industry-relevant problems, getting much-needed practical experience and earning a certificate of competency to support professional growth. As mentioned in [12, 15], this significantly improves the interest of users in the field and encourages more people to accelerate their codes. Furthermore, supervising skilled and interested students across Europe in projects with state-of-art topics using GPUs during PRACE Summer of HPC, brought more interest and possible workforce for the area.

4.2 Quantum Computing

The rise of Quantum Computing (QC) as the next mainstream computing paradigm for code acceleration is gaining momentum in the scientific computing community, promising to change the way we solve real-world challenges. Based on completely different physical rules compared to traditional, classical computing, QC requires a different mind-set and a different approach in the way code is written. This is true due to a number of factors: completely new hardware to be interfaced with classical facilities, classical codes needs to be ported to be suitable to work on quantum hardware and, overall, basic understanding of quantum mechanical rules and how these affects computation. As such, the need for structured and rigorous training for QC is very much needed.

Structured training is a strong challenge for super-computing centres to adapt and prepare materials for emerging technologies such as QC. There are readily-available materials for quantum computing such as:

- *Introductory and Learner level:* Michael Nielsen's book [25] and tutorials [17, 18, 23, 24] are one of the most welcoming when it comes to introduction for QC. As previously mentioned by [20–22], games and interactive environments have significant importance in education and training in HPC, and there are also QC-related games available [7, 32]. There are several very well-prepared materials such as Qiskit textbook [8, 31] Tensorflow Quantum [10], Cirq [13], Pennylane Tutorials [9], however, they are tailored for specific SDKs and hardware. Furthermore, installing a QC environment and using it can be challenging especially for end-users. In addition, SDKs and QC environments are not stable and getting updated regularly.
- *Practitioner level:* Previously mentioned materials also have advanced levels for people who wants to specialise. In addition, there are many articles available and published every day about quantum computing.

In HC, *via* the Explain program we aim at building the next generation of quantum software engineers from bottom up. First, we provide training for the basics of quantum computing with respect to introductory applied quantum mechanics, and afterwards, for specific hardware (e.g., quantum annealing, universal gate-based etc.) and their SDKs. Despite we commonly use universal gate-based systems for hands-on training using simulators, we can tailor the materials according to the specific needs of a project or users. Furthermore, we also encourage and support individuals to take extensive courses from external sources and join related events to be specialists in the area.

5 CONCLUSIONS

In this paper we provide our perspective on how to effectively train industry users, and how to engage with them about wider digital technologies and how these, used efficiently together, can benefit their business. Specifically, we have discussed our three stages education plan. In the first stage, we provide each and every user with a core training on how to use efficiently our HPC system, building a confident and self-sufficient user cohort that can productively use the machine. In the second stage, we engage with industry users building digital innovation awareness. This stage is key to provide businesses with concrete examples of how applied digital strategy can bring benefits to their business. The third and last stage is to provide specialist training tailored to match the business needs of the users, as well as training on novel emerging technologies via

our HNCDI Explain work stream. This allows us to stay competitive meanwhile building the future's work force.

To conclude, through the approaches described in this paper we have introduced businesses to new ways of working, and incorporating new technologies into their research pipeline through enhanced service offerings. Demonstrating the capabilities based on each kind of technology leads to an increased demand to use them. As specific examples, this includes using remote interactive access and visualisation, use of GPUs and quantum computing. Data analysis and AI solutions are also being included in software development work, e.g. using AI models alongside more traditional mathematical models.

ACKNOWLEDGMENTS

This work was supported by the Hartree National Centre for Digital Innovation, a UK Government-funded collaboration between STFC and IBM.

REFERENCES

[1] 2022. The Hartree Centre. https://www.hartree.stfc.ac.uk/.
[2] 2022. Hartree National Centre for Digintal Innovation. https://www.hartree.stfc.ac.uk/Pages/Hartree-National-Centre-for-Digital-Innovation-(HNCDI).aspx.
[3] 2022. HNCDI Explain courses. https://www.eventbrite.com/cc/hartree-centre-explain-training-programme-259399.
[4] 2022. Science and Technology Facilities Council. https://www.ukri.org/councils/stfc/.
[5] 2022. Turbo VNC. https://www.turbovnc.org/.
[6] 2022. UKRI - UK Research and Innovation. https://www.ukri.org/.
[7] TOPTICA Photonics AG. 2022. Quantum Quiz. https://www.toptica.com/quantumquiz
[8] Gadi Aleksandrowicz et al. 2019. Qiskit: An open-source framework for quantum computing. (2019).
[9] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, et al. 2018. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968* (2018).
[10] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J Martinez, Jae Hyeon Yoo, Sergei V Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, et al. 2020. Tensorflow quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989* (2020).
[11] Manuel Caeiro-Rodríguez, Mario Manso-Vázquez, Fernando A Mikic-Fonte, Martín Llamas-Nistal, Manuel J Fernández-Iglesias, Hariklia Tsalapatas, Olivier Heidmann, Carlos Vaz De Carvalho, Triinu Jesmin, Jaanus Terasmaa, et al. 2021. Teaching soft skills in engineering education: An European perspective. *IEEE Access* 9 (2021), 29222–29242.
[12] Xi Chen, Gregory S. Gutmann, and Joe Bungo. 2018. Deep Learning by Doing: The NVIDIA Deep Learning Institute and University Ambassador Program. *CoRR* abs/1812.08671 (2018). arXiv:1812.08671 http://arxiv.org/abs/1812.08671
[13] Cirq Developers. 2022. Cirq. https://doi.org/10.5281/zenodo.6599601 See full list of authors on Github: https://github .com/quantumlib/Cirq/graphs/contributors.
[14] Bence Ferdinandy, Ángel Manuel Guerrero-Higueras, Éva Verderber, Ádám Miklósi, and Vicente Matellán. 2019. Analysis of users' first contact with High-Performance Computing: first approach with ethology researchers. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*. 554–557.
[15] Liqiang He. 2010. Computer architecture education in multicore era: Is the time to change. In *2010 3rd International Conference on Computer Science and Information Technology*, Vol. 9. IEEE, 724–728.
[16] William Humphrey, Andrew Dalke, and Klaus Schulten. 1996. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics* 14 (1996), 33–38.
[17] Andy Matuschak and Michael A Nielsen. 2019. How does the quantum search algorithm work? *URL: https://quantum.country/search* (2019).
[18] Andy Matuschak and Michael A Nielsen. 2019. Quantum computing for the very curious. *URL: https://quantum. country/qcvc (cit. on p. 8)* (2019).
[19] Stefano Mensa. 2022. Scafell Pike Customer Onboarding. https://www.hartree.stfc.ac.uk/Pages/The_Hartre_Centre_Scafell_Pike_Introductory_Manual_V1_6.pdf.
[20] Julia Mullen, Chansup Byun, Vijay Gadepally, Siddharth Samsi, Albert Reuther, and Jeremy Kepner. 2017. Learning by doing, High Performance Computing education in the MOOC era. *J. Parallel and Distrib. Comput.* 105 (2017), 105–115.
[21] Julia Mullen, Lauren Milechin, and Dennis Milechin. 2021. Teaching and learning HPC through serious games. *J. Parallel and Distrib. Comput.* 158 (2021), 115–125.
[22] Julia Mullen, Albert Reuther, William Arcand, Bill Bergeron, David Bestor, Chansup Byun, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Peter Michaleas, Lauren Milechin, Andrew Prout, Antonio Rosa, Siddharth Samsi, Charles Yee, and Jeremy Kepner. 2018. Lessons Learned from a Decade of Providing Interactive, On-Demand High Performance Computing to Scientists and Engineers. In *High Performance Computing*, Rio Yokota, Michèle Weiland, John Shalf, and Sadaf Alam (Eds.). Springer International Publishing, Cham, 655–668.
[23] Michael Nielsen. 2010. Quantum computing for the determined. (2010).
[24] Michael A Nielsen. 2010. The Quantum Computing for the Determined. *URL: https://www.youtube.com/* (2010).
[25] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.
[26] Rajendra K Raj, Carol J Romanowski, Sherif G Aly, Brett A Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I Gordon, Cruz Izu, Shahram Rahimi, et al. 2020. Toward High Performance Computing Education. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. 504–505.
[27] Rajendra K Raj, Carol J Romanowski, John Impagliazzo, Sherif G Aly, Brett A Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I Gordon, Cruz Izu, et al. 2020. High performance computing education: Current challenges and future directions. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. 51–74.
[28] Felipe Augusto Lara Soares, Cristiane Neri Nobre, and Henrique Cota de Freitas. 2019. Parallel programming in computing undergraduate courses: A systematic mapping of the literature. *IEEE Latin America Transactions* 17, 08 (2019), 1371–1381.
[29] Leonardo BA Vasconcelos, Felipe AL Soares, Pedro Henrique MM Penna, Max V Machado, Luís Fabrício W Góes, Carlos Augusto PS Martins, and Henrique C Freitas. 2019. Teaching parallel programming to freshmen in an undergraduate computer science program. In *2019 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.
[30] George Williamson. 2019. Virtual Wind Tunnel. *URL: https://www.hartree.stfc.ac.uk/Pages/Virtual-Wind-Tunnel.aspx* (2019).
[31] James R. Wootton, Francis Harkins, Nicholas T. Bronn, Almudena Carrera Vazquez, Anna Phan, and Abraham T. Asfaw. 2021. Teaching quantum computing with an interactive textbook. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 385–391. https://doi.org/10.1109/QCE52317.2021.00058
[32] Iftact Yakar. 2022. Quantle. https://github.com/deduckproject/quantle.

Sustainable and Scalable Setup for Teaching Big Data Computing

Linh B Ngo
West Chester University of Pennsylvania
West Chester, PA
lngo@wcupa.edu

Hoang Bui
Loyola University Maryland
Baltimore, MD
hdbui@loyola.edu

ABSTRACT

As more students want to pursue a career in big data analytics and data science, big data education has become a focal point in many colleges and universities' curricula. There are many challenges when it comes to teaching and learning big data in a classroom setting. One of the biggest challenges is to prepare big data infrastructure to provide meaningful hands-on experience to students. Setting up necessary distributed computing resource is a delicate act for instructors and system administrators because there is no one size fit all solutions. In this paper, we propose an approach that facilitates the creation of the computing environment on both personal computers and public cloud resources. This combined approach meet different needs and can be used in an educational setting to facilitate different big data learning activities. We discuss and reflect on our experience using these systems in teaching undergraduate and graduate courses.

KEYWORDS

Big Data Computing, Learning Activities, Apache Spark

1 INTRODUCTION

Multicore processors have become standard in modern personal computing devices. Linux Kernel Subsystem and Hypervisor components have ensured Windows-based computers to have access to the same software libraries commonly used in parallel and distributed computing (PDC) education such as *pthreads* [17], *OpenMP* [22] and *OpenMPI* [23]. This enables students to carry out PDC learning activities on their personal computers rather than fully dependent on large-scale computing resources. When it comes to big data computing (BDC) topics, computing environment setup becomes more complex. For example, Apache Spark, a popular big data analytic platform, is not a library to be linked and invoked at run time but a complex ecosystem that needs to be installed, configured, and deployed. Programs are then submitted to this platform for execution. In addition to multicore requirements, available memory and local storage are also critical resources to be managed. To date, BDC education relies mainly on distributed resources with a preference for on-site physical cluster [15].

https://doi.org/10.22369/issn.2153-4136/14/1/7

In this work, we present several deployment varieties for individualized computing environments together with various BDC learning activities. These approaches range from direct installation and configuration on personal computing devices, development of workflow on local clusters to indirect deployment through containerization, and large-scale temporary deployment on federal cloud resources. This provides a sustainable approach to BDC education where the burden of maintain computing resources is not solely placed on academic institutions and students have access to a learning environment beyond the duration of the courses. These approaches help creating and disseminating BDC courses at two academic institutions that lack support for large-scale infrastructures.

The remainder of the paper is structured as follows. Section 2 presents our approach to maintain a sustainable BDC learning environment. Section 3 describes the learning activities and assessments. In Section 4 we discuss our overall classroom experience, including descriptions of previous taught courses, students evaluation and learning outcomes, and the lessons learned. Finally, we conclude our paper and discuss future work in Section 5.

2 SUSTAINABLE AND SCALABLE SOLUTIONS

There are primarily three approaches to providing computing environment to big data education: physical cluster [7], virtual cluster [14], or cloud-based solutions [24]. Given these approaches require institutional investments and extensive technical knowledge, scalability and sustainability can be limited for smaller institutions. These solutions can become limited available to students due to limitation such as computing credits (cloud), on-campus access (physical or virtual), or resource contention (physical or virtual). Students can lose access to resources after the course is ended, hindering the potential of further self studying.

We define sustainable solutions as approaches that do not place significant financial and technical burden on students and academic institutions. A sustainable option for BDC learning environment, therefore, is one that is deployed on students' personal computer (PC). However, it is critical that learning activities behave exactly the same on personal computers or large-scale resources, except for run time performance.

2.1 Infrastructure on personal computers

There are three varieties of deploying Spark on PC: direct deployment, single-node containerized local deployment, and multi-node containerized cluster deployment. Today, a fairly minimal and inexpensive (relative) laptop boasts a dual-core CPU, 4GB of memory, and 32GB of storage. A direct installation of Apache Spark [28]

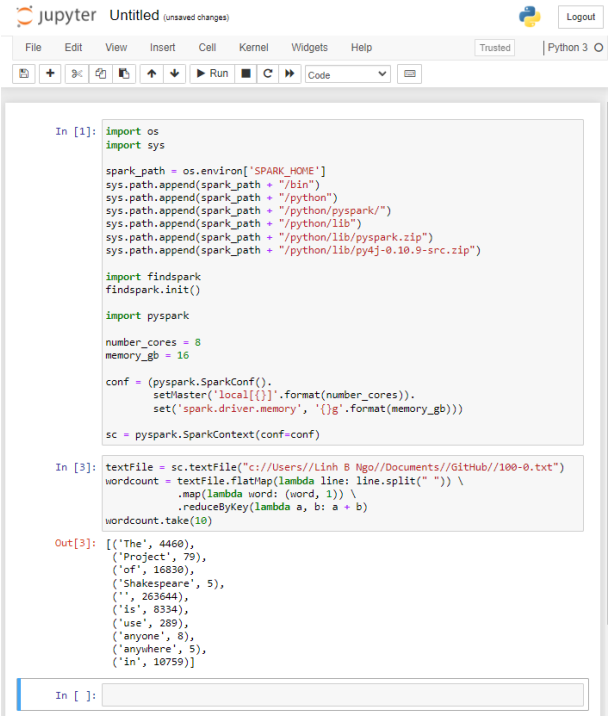


Figure 1: Jupyter notebook with Spark cell setup and Word Count execution

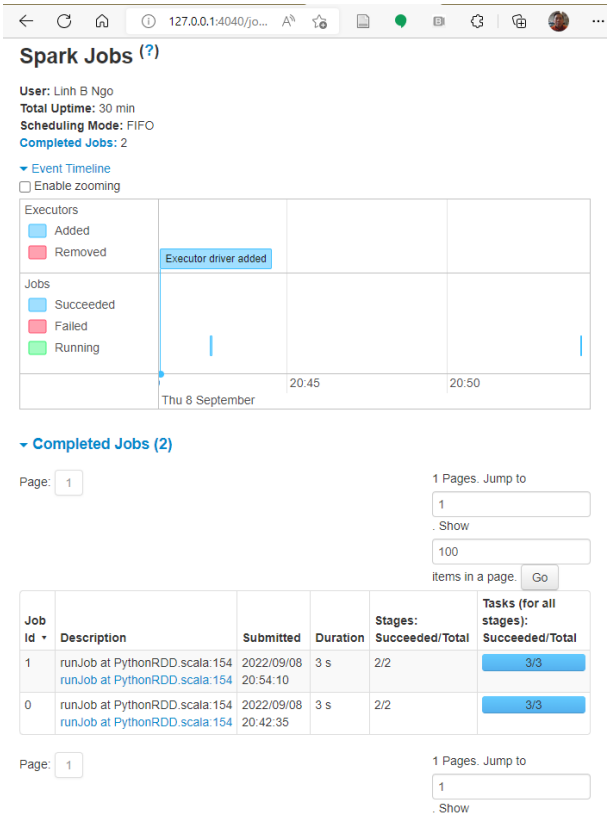


Figure 2: Spark Web UI on local server

and PySpark [13] is necessary to avoid overhead that could happen from containerized solutions. The trade-off in this case is the added complexity of managing various installations in a Windows environment. For more powerful devices, single-instance Docker solutions can be used, where all required libraries for Spark and Python are already configured inside the container. With top-of-the-line PCs, we can also deploy a multi-node Spark cluster that is housed in multiple containers. In all scenarios, we are assuming Windows installation as it is the most popular operating system used by students.

For direct deployment, the following components must be setup: 1) Anaconda [5], 2) Apache Spark, 3) Java, and 4) Hadoop-Windows utilities. Out of these four components, Anaconda potentially takes up the most space (approximately at least 500MB) and requires an installation process. While it is possible to selective pick only relevant Python components necessary to support Apache Spark, the steps will be lengthy and tedious. Students lacking command-line experience and administration skill will likely encounter errors, creating technical overhead inside and outside of the classroom. Java can either be installed or decompressed to a specific location. Apache Spark and Hadoop-Windows utilities need to be downloaded and decompressed to specific locations. Once everything is in place, environment variables need to be set for *ANACONDA_HOME*, *SPARK_HOME*, *HADOOP_HOME*, and their corresponding sub-directories to executable files in *PATH* via Windows' System Properties.

After a Jupyter notebook is created, a block of template code is provided to setup the launching of a local Spark cluster. This template includes getting the location of Spark's installation via *SPARK_HOME* and append relevant supporting libraries to the notebook's Python kernel. Students can specify the size of the cluster via number of cores and amount of memory in GB. Finally, *PySpark* will launch the local Spark cluster. Figure 1 demonstrates the execution of the template cell, and the subsequent running of another cell that runs the word count activities and returns the top ten unique words' counts. Figure 2 shows the records of the submitted Spark jobs on the local 127.0.0.1:4040 address.

In both single-node [2] and multi-node [1] containerized deployments, the key setup step is to install Docker Desktop. The challenge is the enabling of virtualization support on older laptop models via BIOS. This issue has gradually been reduced over time as newer laptops have virtualization enabled by default. The deployments launch the Docker container(s) and expose the default port of the Jupyter notebook server to the host machine, making Jupyter available to students via the host browser. One downside of this approach is the limited access to Spark's Web UI. While it is possible to expose the primary interface of the Web UI, additional log information resides on individual Spark worker's container whose port must be exposed separately. It is possible to examine the log from the terminal using *docker log* command. However, this creates a potential point of failure/technical overhead for students. An example multi-node containerized deployment is shown in Figure 3.

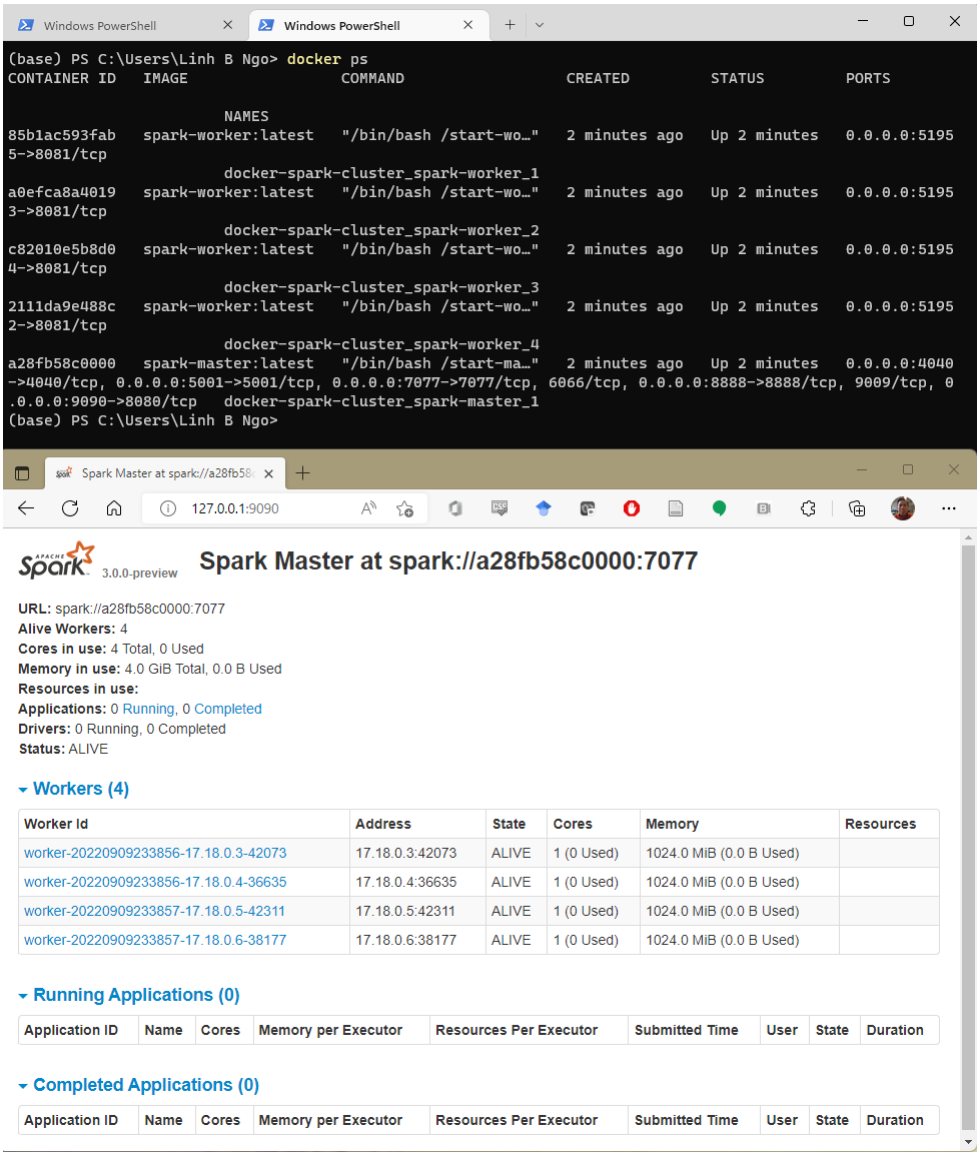


Figure 3: Multi-node containerized deployment of Spark cluster

2.2 Scaling to community cloud

The containerized solutions for single and multi-node Spark cluster can also be used to deploy at scale on CloudLab, a federal cloud resource that is available for research and education purposes [21]. A CloudLab experiment needs to be deployed prior to class. This experiment launches a single Docker node or a multi-node Docker Swarm [3]. Students can launch the containerized deployments here on the experiment and access the Jupyter server via the public IP address of the experiment's head node.

As CloudLab is designed to be an experimental test-bed, cloud allocations are provisioned within limited timing duration (16 hours) that are not suitable for sustained learning activities. However, long-term availability of personal computing devices can be combined with CloudLab to create a learning model that enables students to build their big data workflow locally using smaller data set and test their solutions on CloudLab using larger data sets.

The combination of personal computing devices and public cloud resources facilitates sustainable and scalable solutions to provide learning environments for BDC topics. In the next section, we will discuss learning activities that are created to support this approach.

3 LEARNING ACTIVITIES AND ASSESSMENT

While designing learning activities, we focus on guiding students through fundamental steps in the big data processing pipeline. These steps include identifying data sources, acquiring and ingesting raw data, and analyzing curated data. Students are exposed to both the underlying theory behind big data techniques and the software tools and infrastructures that implement and support these

techniques. Assessments include both short-term assignment and a semester-long data analysis project.

3.1 Learning activities

Data acquisition: The first topic of interest is data identification, acquisition, and curation. The experience of combing through a large number of data sets can be rewarding but also can be frustrating at times. For this topic, students learn to narrow down their topic area and not to focus solely on quantity but on quality of the data sets being examined. Qualities such as cleanliness, reliability and uniqueness have a direct impact on subsequent learning activities. There are many publicly available data sets spanning across many topic areas for students to explore. Examples include *data.gov* [10] for data related to government, climate, health, energy, and economy, *Kaggle* [16] for health, science, sports, crypto, and entertainment data, *UCIML* [26] for science and engineering data, *Nasdaq Datalink* [20] for financial and business related data. Table 1 shows a list of sources and topic areas for publicly available data sets.

Additionally, we also make security data collected from our own Linux servers available to students. For larger data sets, smaller samples that can easily be processed on PCs are generated.

Programming models: Existing big data toolkits (e.g., Hadoop [12], Spark [28], AllPairs [19], DataSpaces[25], etc) already provide an extensive collection of ready-to-use functionalities. It is critical that students understand the underlying programming paradigms implemented in these functionalities. They are to momentarily step away from the traditional procedural and object-oriented paradigms where functions and objects are the targets of programming activities. Instead, they are to focus on the data pipelines and how these pipelines will eventually produce the desired results. The MapReduce programming paradigm [11] is one such dataflow programming paradigm, where programmers utilize 'map' and 'reduce' functions to form the data pipelines. This paradigm is implemented in Hadoop MapReduce, Apache Spark, and many other big data frameworks.

3.2 Assessments

In addition to the standard quizzes and exams that assess students' on their understanding of foundational concepts, assignments and semester projects are key components to the assessment process. The following assignments and projects were disseminated to students and carried out primarily using the infrastructures described in Section 2.

Assignments: Assignments are used for WCUPA's BDC courses. There is a total of five assignments that work on progressively bigger and more complex data sets. Students demonstrate their understanding by implementing well-known algorithms such as PageRank and K-mean clustering using MapReduce programming paradigms and apply them on the data.

- Assignment 1 is a straightforward demonstration that students are able to deploy Spark on their PC/laptops. This assignment requires students to provide a series of screenshots showing working Jupyter notebooks, Spark WebUI, and success WordCount results. The assignment serves as a confirmation that all students have access to adequate infrastructure to continue the course.
- Assignment 2 provides students with an actual security log of a public-facing computer. Students are to study the log and provide answers to the following questions: 1) How many failed access attempts? 2) Which countries these attempts are generated from? 3) What are the attempted usernames? 4) Which date has the highest attack frequencies? These questions require students to become familiar with Spark's actions and transformations and also to learn how to examine complex textual data.
- In Assignment 3, students are first introduced to a *big* data set: the user information portion of Yelp's academic data set [4]. This data set is approximately 1.8Gb compressed, which is large enough to be inconvenient. Besides descriptive statistics, students are required to identify the top ten influential users from this data set. This particular requirement is open-ended, as students will need to justify their choice of attributes that define level of influence.
- Assignment 4 is an extension of assignment 3, where students now use all data within the Yelp data set (user, review, and business) to study characteristics of the influential users and identify their pattern of restaurant visits (local, regional, or east-west coasts). This assignment is where students can decide to apply complex techniques such as PageRank and K-mean clustering.
- Assignment 5 introduces students to Kaggle [6]. The assignment involves two parts. In part 1, students are to participate in the introductory "Titanic - Machine Learning from Disaster" competition but use Spark and its supporting libraries to carry out the prediction task. In part 2, students study the cryptocurrency data set on Kaggle. While this data set is not overly large (approximately 300Mb), the text line themselves contains non-standard characters and are not easily tokenized.

Semester Project: We have students design and implement a data analysis workflow to analyze the data set of their own choice. The goal of this activity is to get students ready to apply what they learn to work on real-world problem beyond the classroom. We ask students to come up with at least five interesting questions they want to answer from the chosen data set. If the student decides MapReduce programming model is suitable to answer those questions, the student would write their own mapper and reducer functions specific to each question. A prototype of the workflow is developed on small sampled data and run on PCs. Later, this prototype is migrated to run on a large-scale BDC environment that could be on-site or cloud-based, depending on the institution' resources. We evaluate the entire workflow by processing different data sizes and scaling out across different number of computers and whether the results support answering the proposed questions.

Case Study: One of our students wanted to perform a study of educational data, specifically,identifying the correlation between a student's gender, course work performance and the likeliness of he/she pursuing a career in STEM after high school graduation. The student looked into a number of educational data sets from the National Center for Education Statistics (NCSE) and chose a

Table 1: Popular sources for data sets

Source	Topic Areas
data.gov	Government, Climate, Health, Energy, Economy,...
Kaggle	Health, Science, Sports, Crypto, Entertainment,...
UCI ML Repository	Life Science, Physical Science, Engineering,...
Nasdaq Datalink	Financial, Real Estate, Banking,...

data set from the High School Longitudinal Study[27]. The study surveyed over 20,000 students from more than 900 both public and private schools.

After obtaining a data set, the next step was for the student to propose a list of questions they want to investigate. The questions are listed in Table 3. As an example of the type of insight student could derive from the data set, he/she propose a hypothesis which state that a student with higher GPA is more likely to take post-secondary classes (college/university) because GPA is a good indication of the student's future academic aspiration in higher education. The student then proceed with creating customized mapper and reducer functions to attempt to validate the hypothesis. Figure 4 shows a part of a mapper code.

```
if(gpaNum == -10) {
    gpaRange = "error␣parsing";
}
else if(gpaNum < 0 && gpaNum != -10){
        gpaRange = "missing␣GPA";
} else if (gpaNum >= 0 && gpaNum < 0.5 ){
        gpaRange = "0.0-0.5";
}else if (gpaNum >= 0.5 && gpaNum < 1.0 ){
        gpaRange = "0.5-1.0";
} else if(gpaNum >= 1.0 && gpaNum < 1.5){
        gpaRange = "1.0-1.5";
} else if(gpaNum >= 1.5 && gpaNum < 2.0){
        gpaRange = "1.5-2.0";
}else if(gpaNum >= 2.0 && gpaNum <2.5){
    gpaRange = "2.0-2.5";
...
}
```

Figure 4: A snippet of a customize mapper function

The analysis result validated the hypothesis. Table 2 shows as the GPA increases, there are more student answered yes to indicate they were talking postsecondary courses after graduating from high school.

4 DISCUSSION

The deployment varieties on PC presented in Section 2 have been used in one BDC course from West Chester University of Pennsylvania (WCUPA) and a series of independent study course from Western Illinois University (WIU). Both institutions are regional public universities and lack either infrastructure (WCUPA) or personnel support (WIU) to deploy and maintain large-scale computing infrastructures for regular BDC courses. The adoption of the above-mentioned deployment varieties enable teaching and learning activities of BDC topics and subsequent course creation.

Table 2: Student response to the questions about taking post-secondary courses

GPA	Yes	No	Don't know
0.0-0.5	32	58	49
0.5-1.0	70	158	110
1.0-1.5	195	343	189
1.5-2.0	590	618	319
2.0-2.5	1299	708	329
2.5-3.0	2073	617	269
3.0-3.5	2449	417	169
3.5-4.0	2768	202	70
4.0+	3390	99	16

4.1 Course descriptions

At WCUPA, there was no BDC course prior to 2019. A special topic course on complex large-scale systems was offered with emphasis on BDC contents during the Winter 2019 semester. During this initial offering, the multi-node containerized local deployments were introduced to the course. At this time, Docker required Docker Toolkit and VirtualBox to support containerization for older versions of Windows and Mac, and several students with much older machines could not run a multi-node solution. A single-node solution was developed and introduced as an alternative. After the initial offering in Winter 2019, the course was offered once again as a special topic course in Winter 2020. During this semester, the direct deployment approach was introduced to students. Instructions for Docker-based deployments were made available to students as alternatives but were not formally introduced in class. Beginning Fall 2021, the course was converted into a regular Fall-semester course titled "Big Data Engineering."

Because traditional BDC courses are not offered at WIU, students who want to gain experience working with big data often do so through an independent study course. The course includes most of the activities outlined in Section 3 culminating with a semester long final project. This specific independent study course has been taken by WIU graduate students four times in recent years.

4.2 Student outcomes and feedback

Descriptive summary of student enrollments during for BDC course offering at WCUPA is presented in Table 4. It should be noted that for the Fall 2022 semester, class size was capped at 35 due to limited physical seating. In general, students' verbal feedback has been positive. For Winter semesters, there were no formal evaluation

Table 3: Proposed questions by a student [18]

#	Questions
1	Does a higher high school GPA correlate to enrollment in a post-secondary education program towards earning a bachelor's?
2	Does a higher high school GPA correlate to the number of STEM courses enrolled in during high school?
3	Is there a correlation between gender and the decision to enroll in a post-secondary educational program?
4	Is there a correlation between gender and the amount of STEM courses taken during high school?
5	Does a positive response about the value of math also correlate with the number of science related courses enrolled in during high school?
6	How do different genders perceive the importance of math and their own ability to do well in math compared to that of the other gender?

Table 4: BDC Course offerings and enrollments at WCUPA.

Semester	Enrollment
Winter 2019	13
Winter 2020	20
Fall 2021	40
Fall 2022	35

process. However, in-class interactions had been positive and no significant technical issues happened. During Winter 2020, two students decided to follow up on the security analysis assignment and expanded the work to cover all system logs of department computers. This resulted in a publication that won Best Student Paper award at a regional conference [9]. For the first regular offering, students' ratings of course contents have been well-above departmental, college, and university's mean rating. As Fall 2021 was the first semester back after Covid-19, no verbal feedback from students was made available.

4.3 Lesson learned

We offer the following lessons to summarize our experience in teaching big data computing to both undergraduate and graduate students. We hope these can be applied to sustain and improve teaching and learning experience for fellow educators and students in the future.

Keep sustainability and reproducibility at the forefront: Limited access to workable resources is the primary obstacle to teaching and learning BDC, or PDC for that matter. Helping students to deploy a suitable computing environment on their PCs ensures access and allows students to at least learn about the underlying theories, even without scalability demonstration.

Give students some autonomy: Having access to the computing environment on their own PCs will also contribute to students' autonomy in learning and experimenting with various data sets from topics that they are interested in. Many data sources available on 'data.gov' are actually small in size (dozens of megabytes) and well suited for PC processing. Working with data sets in topic areas that they can relate to either their future career or their personal hobbies can help boost students' motivation.

Be ready to clean data: Ideally, students are able to find a useful and clean data set that contains all relevant information. However, this is almost always not the case, and students should be reminded to be mindful about the cleanliness and trustworthiness of their collected data. Data cleaning is an important part of the data analysis workflow. Additional data sets might be needed before the final data product is ready to be analyzed.

Start small, then scale-up: Students can create a small sample from a big data set while maintaining original statistical properties. They would then design, implement, and deploy an end-to-end analysis prototype workflow for the small sample on their PC environment. A scale-up deployment on the large-scale resource is carried out later. One of the side advantage of starting small is failing faster. If something go wrong, students would have chances to alter their design and implementation and rerun their experiments. Once they are confident on the correctness of their analysis, they can deploy to the large-scale resource for final validation and performance evaluation.

Use high level abstractions to speed up progress: Although we spend time on low level programming paradigms to provide necessary foundational knowledge on BDC, modern frameworks such as Apache Hadoop or Apache Spark provide students with more suitable functionalities to build their data analysis workflow. It is important for students' learning growth that they can step away from the low-level paradigms and focus on designing an appropriate dataflow pipeline. The question then becomes, what functionalities would they need to *shape* their pipeline. In doing so, students can leave complex decisions such as coordination and synchronization among multiple compute nodes to the framework.

5 CONCLUSION AND FUTURE WORK

In this paper, we present our approach to create a sustainable and scalable approaches in setup personal computing environment for big data computing. Our approach potentially can free educators from tedious tasks of maintaining distributed computing infrastructure. Instead, they can focus on teaching and mentoring activities. For students, they will have access to resources that can be recreated and duplicated outside classrooms, enabling self-learning beyond the scope and duration of the class. We also discuss a list of active learning activities that play a large role in achieving many important learning objectives.

With the gradual changes in Windows' toward supporting Linux environment and better yet inexpensive computers, additional work needs to be done to continue improving the above approaches. These include, but not limited to

- Creating better documentation for the direct deployment process. Video instructions might be more useful than static documentation.
- Convert the multi-node Docker deployment from using Docker Compose to using Kubernetes [8]. This allows this approach

to be deployed as a centralized infrastructure if a cloud resource becomes available.
- Improve the Docker deployment to make external data and code integration more dynamic. This will reduce complexity of importing/uploading external materials into the containerized environment.

REFERENCES

[1] 2022. Docker compose file for Multi-node Spark Server. https://github.com/linhbngo/docker-spark-cluster
[2] 2022. Docker image for Single-node Spark Server. https://github.com/linhbngo/docker-images/tree/master/csc467
[3] 2022. Docker profile to launch Docker and Docker Swarm. https://github.com/CSC468-WCU/csc468cloud/tree/docker
[4] 2022. Yelp Open Dataset. https://www.yelp.com/dataset
[5] Anaconda. [n. d.]. Anaconda: Open-source Python distribution platform.
[6] Casper Solheim Bojer and Jens Peder Meldgaard. 2021. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting* 37, 2 (2021), 587–603.
[7] Richard A Brown. 2009. Hadoop at home: large-scale computing at a small college. In *Proceedings of the 40th ACM technical symposium on Computer science education*. 106–110.
[8] Brendan Burns, Joe Beda, Kelsey Hightower, and Lachlan Evenson. 2022. *Kubernetes: up and running*. " O'Reilly Media, Inc.".
[9] Tyler Clark, Kevin Codd, and Linh B. Ngo. 2021. Studying break-in attempts across multiple servers using Apache Spark and security logs. In *IFIP International Conference on Network and Parallel Computing*. Annual Spring Conference of the Pennsylvania Computer and Information Science Educators.
[10] data.gov. 2022. The home of the U.S. Government's open data. https://www.data.gov
[11] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.
[12] Jens Dittrich and Jorge-Arnulfo Quiané-Ruiz. 2012. Efficient big data processing in Hadoop MapReduce. *Proceedings of the VLDB Endowment* 5, 12 (2012), 2014–2015.
[13] Tomasz Drabas and Denny Lee. 2017. *Learning PySpark*. Packt Publishing Ltd.
[14] Joshua Eckroth. 2016. Teaching big data with a virtual cluster. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 175–180.
[15] Jesse Eickholt and Sharad Shrestha. 2017. Teaching big data and cloud computing with a physical cluster. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 177–181.
[16] Kaggle. 2022. Kaggle Open Datasets. https://www.kaggle.com/datasets
[17] Henry Kasim, Verdi March, Rita Zhang, and Simon See. 2008. Survey on parallel programming model. In *IFIP International Conference on Network and Parallel Computing*. Springer, 266–275.
[18] Corbett Megan. 2016. Utilizing the Apache Hadoop MapReduce to Analyze Trends in High School Students' Performance and Postsecondary Decisions. *Independent Study Report* (2016).
[19] Christopher Moretti, Hoang Bui, Karen Hollingsworth, Brandon Rich, Patrick Flynn, and Douglas Thain. 2009. All-pairs: An abstraction for data-intensive computing on campus grids. *IEEE Transactions on Parallel and Distributed Systems* 21, 1 (2009), 33–46.
[20] Nasdaq. 2022. Nasdaq Datalink. https://data.nasdaq.com/
[21] Linh B Ngo and Jeff Denton. 2019. Using CloudLab as a Scalable Platform for Teaching Cluster Computing Ambassador Program. *The Journal of Computational Science Education* 10, 1 (2019).
[22] OpenMP. 2022. The OpenMP API specification for parallel programming. https://www.openmp.org.
[23] OpenMPI. 2022. A High Performance Message Passing Library. https://www.open-mpi.org
[24] Ariel S Rabkin, Charles Reiss, Randy Katz, and David Patterson. 2012. Experiences teaching MapReduce in the cloud. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 601–606.
[25] Melissa Romanus, Fan Zhang, Tong Jin, Qian Sun, Hoang Bui, Manish Parashar, Jong Choi, Saloman Janhunen, Robert Hager, Scott Klasky, et al. 2016. Persistent data staging services for data intensive in-situ scientific workflows. In *Proceedings of the ACM International Workshop on Data-Intensive Distributed Computing*. 37–44.
[26] UCI. 2022. UCI ML Repository. https://archive.ics.uci.edu/ml/index.php
[27] USDE-NCES. 2016. The School Longitudinal Study, 2009-2013 United States.
[28] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. 2016. Apache spark: a unified engine for big data processing. *Commun. ACM* 59, 11 (2016), 56–65.

Exascale Computing Project's Broadening Participation Initiative

Suzanne Parete-Koon
Oak Ridge National Laboratory
Oak Ridge, Tennessee
paretekoonst@ornl.gov

Mary Ann Leung
Sustainable Horizons Institute
Rancho Mirage, California
mleung@shinstitute.org

Sreeranjani Ramprakash
Argonne National Laboratory
Argonne, Illinois
jini@alcf.anl.gov

Lois Curfman McInnes
Argonne National Laboratory
Argonne, Illinois
curfman@mcs.anl.gov

ABSTRACT

The mission of the U.S. Department of Energy's (DOE) Exascale Computing Project (ECP; https://www.exascaleproject.org) is to ensure all the necessary pieces are in place for the nation's first exascale systems. The project is delivering an ecosystem that includes mission critical applications and an integrated software stack, while working closely with U.S. high-performance computing (HPC) hardware companies to identify and drive the development of advanced computer system engineering and hardware components. All of these elements are necessary to enable fully functional, capable exascale computing environments, which are critical to national security, scientific discovery, and a strong U.S. economy. ECP is composed of hundreds of researchers and engineers from various DOE national laboratories as well as academic and industry partners. [1]

This article gives an overview of ECP's Broadening Participation Initiative (https://www.exascaleproject.org/hpc-workforce/), which has the mission of establishing a sustainable plan to recruit and retain a diverse workforce in the DOE high-performance computing community by fostering a supportive and inclusive culture within the computing sciences at DOE national laboratories. We will describe key activities within three complementary thrusts: establishing an HPC Workforce Development and Retention Action Group, creating accessible 'Intro to HPC' training materials, and launching the Sustainable Research Pathways for High-Performance Computing (SRP-HPC) workforce development program. We are leveraging ECP's unique multilab partnership to work toward sustainable collaboration across the DOE community, with the long-term goal of changing the culture and demographic profile of DOE computing sciences.

KEYWORDS

High Performance Computing, Education, Diversity, Equity, Inclusion, Workforce Development

1 HPC WORKFORCE DEVELOPMENT AND RETENTION ACTION GROUP

The mission of the HPC Workforce Development and Retention (HPC-WDR) Action Group is to enable DOE national laboratories and their related computing communities to share their collective insight for inclusive and equitable workforce development and retention for high-performance computing. Representatives from various national laboratories and associated universities meet regularly to share ideas, catalog best practices, and develop recommendations and strategies for improvement.

The first two HPC-WDR projects are a webinar series and a website focused on best practices for developing a diverse, equitable, and inclusive HPC workforce culture. Webinars (https://www.exascaleproject.org/workforce-development-seminar-series/) have been held on best practices in mentoring and how to normalize inclusion by embracing our differences. The most recent webinar covered how to be a good workplace ally. The speakers are drawn from the HPC community. The website, once developed, will host an archive of webinar recordings, along with information on workforce and cultural development opportunities and best practices drawn from the participating computing communities.

2 INTRO TO HPC

The mission of the Intro to HPC team is to provide accessible introductory material to HPC, thereby addressing gaps in, and expanding the pipeline of, people with foundational HPC skills. The first target is the development of an intensive HPC/AI course aimed at advanced undergraduate students and early graduate students in underrepresented groups. The team is working collaboratively across DOE national laboratories and communities to develop a curriculum, including hands-on HPC exercises. The team has issued a broad call for interest and identified potential contributors from across the ECP and national laboratory staff. To determine a plan for the program, they are leveraging experience and a framework

[1] Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

https://doi.org/10.22369/issn.2153-4136/14/1/8

from Argonne National Lab's Education Department. Listening sessions with a subset of the computational postdoctoral population at Argonne and Oak Ridge National Labs were held to uncover HPC topics that were (1) useful and (2) missing from their undergraduate and early graduate education. Once the curriculum is complete, the plan is to have course materials freely available online for community use.

The Intro to HPC team is also working with universities by conducting listening sessions to understand the challenges of teaching advanced computing topics and how to address them. The team is focusing on minority-serving institutions in the U.S. that offer 4-year degrees with computer science or related departments. The first listening session was completed on April 21, 2022. Invitations were sent to 40 institutions, resulting in participation of 14 professors from 12 institutions. A listening session report including lessons learned is now available. Ultimately, the team plans to work with administrators and faculty at interested universities to develop and implement Intro to HPC programs at their institutions.

3 SUSTAINABLE RESEARCH PATHWAYS FOR HPC

Sustainable Research Pathways for HPC (SRP-HPC; https://shinstitute.org/srp-hpc/) is an inclusive workforce development program that began this year with a cohort of 61 students from underrepresented groups in HPC and related faculty. They are working with ECP teams at 9 DOE labs on a variety of projects across application development, software technologies and advanced computing facilities. The program includes onboarding at the ECP Annual meeting and a 10-week summer experience that incorporates extended opportunities for mentoring and community building. In addition to boosting participants' careers by giving them the opportunity to explore cutting-edge research opportunities at DOE labs, the program also focuses on helping people learn how to work together and unlearn biases so that inclusion becomes a normal practice.

The SRP-HPC program is based on a program started in 2015 at Lawrence Berkeley National Laboratory (Berkeley Lab) that was developed by the Sustainable Horizons Institute, a 501(c)3 nonprofit dedicated to building inclusive scientific communities. The ECP Broadening Participation Initiative has scaled up the SRP concept across the ECP community.

4 CONCLUSION

Through these three complementary thrusts, the Exascale Computing Project's Broadening Participation Initiative is helping to build a more diverse workforce and foster an inclusive professional environment for high-performance computing through the national labs and their related academic partners. Submission and reviewing guidelines, and methodology: http://submissions.supercomputing.org/reproducibility

ACKNOWLEDGMENTS

This work was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative and by the resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725 and by the resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Computational Analysis of SARS-CoV-2 Therapeutics Development

Samuel Biggerstaff
Department of Chemistry
Centre College
Danville, KY

Jennifer L. Muzyka
Department of Chemistry
Centre College
Danville, KY
jennifer.muzyka@centre.edu

David Toth
Department of Computer Science
Centre College
Danville, KY
david.toth@centre.edu

ABSTRACT

SARS-CoV-2 (also known as COVID-19) is a coronavirus that has recently emerged and impacted nearly every human on the planet. The nonstructural protein 12 (NSP 12) is an RNA-dependent RNA polymerase that replicates viral RNA in a cell to infect it. Interrupting this function should prohibit the virus from replicating within the body and would decrease the severity of the virus's effects in patients. The objective of this project is to identify potential inhibitors for NSP 12 that might be suitable as antiviral drugs. Thus, we obtained the structure of NSP 12 from RCSB's protein data bank. The protein structure was analyzed using computer software (Chimera and PyRx), and ligands obtained from the ZINC database and RCSB's protein data bank were docked to NSP 12. The resulting binding affinities were recorded, and binding geometries analyzed.

KEYWORDS

Virtual Screening, Drug Discovery, AutoDock Vina

1 INTRODUCTION

Since the end of 2019, the virus SARS-CoV-2, also known as Covid 19, has permeated throughout the cultural, political, and medical fields of nearly every country. The emergence of this virus has altered the day-to-day life of many as they attempt to avoid being infected by SARS-CoV-2. As a result, chemists, biochemists, biologists, and other medical scientists have directed their attention to SARS-CoV-2, its composition, effects, and treatments. Therapeutic treatments are currently of particular interest to the medical community, and two drugs, Molnupiravir and PF-07321332, show promising inhibitory effects against SARS-CoV-2 [12,13]. Additionally, Remdesivir, the only drug currently approved by the FDA for the treatment of SARS-CoV-2, is not as effective as desired [4]. As such, there is a great need for the further development of drugs that would inhibit SARS-CoV-2.

SARS-CoV-2 contains a variety of nonstructural proteins (NSP), each exhibiting their own form and function. These proteins, which are observed on the inside of the host cell, mediate the seven steps of viral replication [17]. Most of these proteins are essential for viral replication. Specifically, several of the 16 NSPs are exceptional drug targets. Protein targets were evaluated on necessity for the virus to replicate, uniqueness of structure from host cell proteins, and conservation of protein sequence for SARS-CoV to SARS-CoV-2. Assessment of the differences between the SARS-CoV-2 protein and host cell proteins is to help reduce side effects. Targeting a viral protein that has a similar structure to the host protein will result in high IC50 values and limited effectiveness. The basis for using the conservation of protein sequence between SARS-CoV to SARS-CoV-2 is that a protein that is mutating quickly will not be a good drug target because mutations can affect drug affinity and binding. Conservation between SARS-CoV to SARS-CoV-2 does not guarantee that there won't be mutations in the active site of the protein that will change binding affinity. It does give a better chance that there won't be a random mutation at any point, including the active site.

Our selected target is the nonstructural protein 12, or NSP 12. This NSP is the RNA-dependent RNA polymerase, meaning that NSP 12 uses RNA as a template to replicate the genome of SARS-CoV-2. Inhibiting NSP 12 would decrease the replication rate, reducing the symptoms of SARS-CoV-2 [4]. The NSPs of SARS-CoV-2 serve as good inhibition targets for therapeutics because of their significant roles in the function of the virus [14]. The NSPs in different variants of SARS-CoV-2 do not differ significantly, so drugs targeting the NSPs of SARS-CoV-2 will inhibit significant functions and will be effective across all observed variants [18]. More significantly, NSP 12 is highly conserved between SARS-CoV and SARS-CoV-2 [10]; thus, it is likely that inhibitors for SARS-CoV-2 will be effective for other coronaviruses [19].

Currently, few drugs have been proven to be successful in inhibiting this nonstructural protein. This is, in part, because there are millions of small molecules that could potentially be used as drugs. Deciding which is the best through experimentation alone would be an extremely long task that would not satisfy the urgent need for SARS-CoV-2 therapeutics. A solution to this dilemma is to take advantage of virtual screening to narrow down the list in a shorter period before beginning experimental trials. One computational program which aids in drug discovery is AutoDock Vina, an open-source program for molecular docking [16]. AutoDock Vina calculates binding affinity between proteins and small molecules in kilocalories per mol (kcal/mol) with a larger negative number indicating a greater binding affinity. For reference, Remdesivir is a nucleotide analog with a binding affinity of -7.8 kcal/mol in our calculations. Remdesivir is a delayed chain terminator that blocks transcription [4]. Since Remdesivir has been shown to be effective in treating patients infected with SARS-CoV-2, any small molecule with a stronger binding affinity might be at least somewhat effective in the treatment of SARS-CoV-2.

2 METHODS

First, the structure of the NSP 12 of SARS-CoV-2 (6YYT) was obtained from the RSCB Protein Data Bank and the structural file was analyzed and optimized in Chimera [4,11].

DOI: https://doi.org/10.22369/issn.2153-4136/14/1/9

The protein NSP12 crystal structure was selected from the Protein Data Bank (PDB). The structure selected was reported by Mariano et al. [9]. The structure selected also had NSP7 and NSP8 bound in the crystal structure. For modeling to get the protein small enough to be functionally useful for the software, NSP7 and NSP8 were manually removed. The ligands to be tested were selectively chosen from the PDB database. Ligands that were nucleotide triphosphates (NTP) or nucleotide monophosphate (NMP) derivatives were chosen for testing. The selected ligand file was downloaded as an SDF file from the PDB website.

Then, two studies were conducted: the targeted study and the general study. In the targeted study, small molecules with a similar structure to RNA were chosen and docked to NSP 12. The bounding box, which tells the docking software where the ligand should be placed, was determined by locating the binding site for RNA on NSP 12. PyRx, a front-end interface for AutoDock Vina, was used to dock these molecules [2]. The chosen molecules were either nucleotides or nucleotide derivatives because NSP 12 typically binds to nucleotides in the body. In the general study, small molecules from the ZINC Database were collected and docked. We downloaded .gz files containing multiple compounds from the ZINC database [5,20]. Then, we wrote a series of Python programs to automate the process [15]. The programs unzipped all the downloaded files, producing a series of text files, each containing multiple compounds. The programs then split the text files into the individual pdbqt files. Next, the programs assembled a new commands text file with one command to run AutoDock Vina per compound. The screening was then started in parallel on a multi-CPU server using xjobs [8]. When there were power failures at various points during the screening, another Python program was run to rebuild the commands text file without the compounds that had already been screened and run it again, so the screen could resume where it had left off. When the screening was complete, a final program was used to assemble a text file with all the compounds and their scores and sorted using the Linux sort command to produce a new file with all the compounds and their scores in the order from the best to worst.

3 RESULTS AND DISCUSSION

3.1 Targeted Study

In the targeted study, 45 nucleotide or nucleotide derivatives were docked to NSP 12 using PyRx, with many molecules demonstrating some compatibility. Of those results, 17 ligands had binding energies more negative than the value calculated for Remdesivir (-7.8 kcal/mol) and 2 met our desired target of -9 kcal/mol (more negative values indicate stronger binding). The best binding ligand was 7-methyl-guanosine-5'-triphosphate-5'-(2'-o-methyl)-adenosine (**V9G**) with a binding affinity of -9.1 kcal/mol (Figure 1). All of the ligands analyzed in the targeted study were nucleotides or nucleotide derivatives, so nearly all of the ligands contained a phosphate group, a 5-carbon sugar, and a nitrogenous base. As a result, the differences in conformations and additional atoms can be analyzed.

For instance, V9G and GTA are very similar in structure (Figure 2). The only difference between the two structures is the presence of an ether or alcohol. In V9G, there is an ether in the place of GTA's alcohol suggesting that a stronger electrostatic negative charge provided by the alcohol group in that location is detrimental to the binding affinity of the ligand. This accounts for a difference of 0.5 kcal/mol, resulting in a significant difference in binding affinity.

Most of the best ligands from the targeted study contained a triphosphate group, and six of the top ten ligands exhibited structures that were similar to those of V9G and GTA with minor differences. The length of the phosphate group may be important in the inhibition of NSP 12 because it creates a molecule of the appropriate size to fit into the active site.

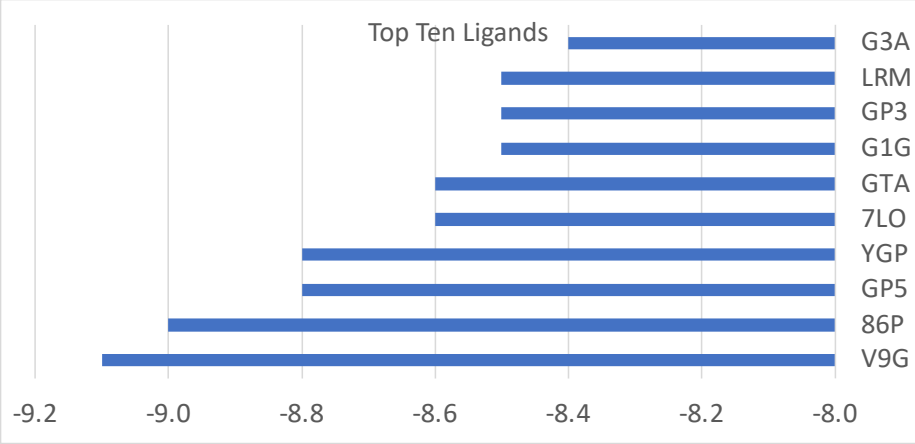


Figure 1. The top ten ligands with the best binding affinity to NSP-12 of SARS-CoV-2 calculated in PyRx from the targeted study.

3.2 General Study

10,582,294 molecules were screened using AutoDock Vina. Of them, about 3,000 were above the desired threshold for being desirable drugs in the inhibition of NSP 12. The best ligand was ZINC000004783172 with a binding affinity of -11.6 kcal/mol.

NSP 12 binds to RNA in the active site. As a result, it is expected that nucleotides and nucleotide derivatives are the ligands that would bind best. Surprisingly, very few of the top-ranking ligands in the general study had characteristics of a nucleotide, and the non-nucleotide molecules of the general study have stronger binding affinities than the nucleotides of the targeted study. There were no phosphate groups in any of the top-ranking ligands, as shown in Table 1. However, when considering that binding affinity is determined by many intermolecular forces such as electrostatic interactions, Van der Waals forces, and hydrogen bonding, small molecules could strongly bind to portions of the protein differently than the nucleotides that are normally found in the binding site.

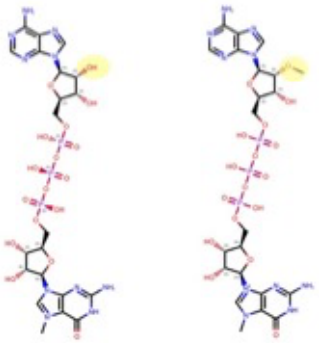
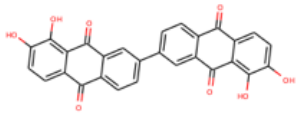
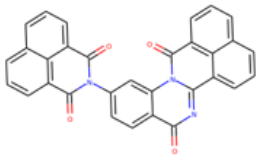
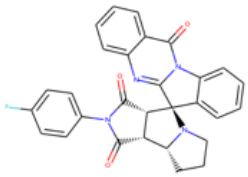
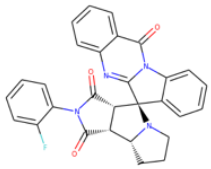
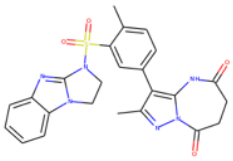
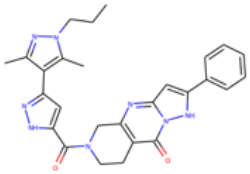
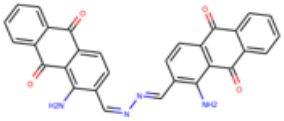
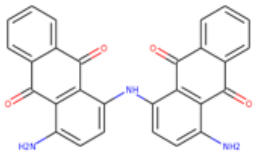
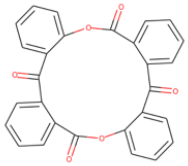


Figure 2. The molecular structures of the ligands GTA and V9G respectively, with their highlighted difference.

The ligand that bound the best, ZINC000004783172, can also be referred to as 7,7'-Bializarin. This molecule is currently being studied as an antibiotic.[20] If this molecule continues to show promising results, then it may be one of the best options for a future therapeutic. According to the calculations from AutoDock Vina, 7,7'-Bializarin should span the entire active site of NSP 12 (Figure 3). This is additionally promising as it ensures that the RNA does not have a location to bind to within the protein's active site, and therefore, cannot replicate.

Table 1. The top nine ligands with the best binding affinity to NSP-12 of SARS-CoV-2 calculated in Autodock Vina from the final general study.

ZINC ID, Binding affinity (kcal/mol)	Structure	ZINC ID, Binding affinity (kcal/mol)	Structure
ZINC000004783172 -11.6		ZINC000004015296 -11.4	
ZINC000035385140 -11.3		ZINC000101500434 -11.2	
ZINC000033122972 -11.2		ZINC000097137247 -11.2	
ZINC000004701175 -11.2		ZINC000003861401 -11.2	
ZINC000004974498 -11.2			

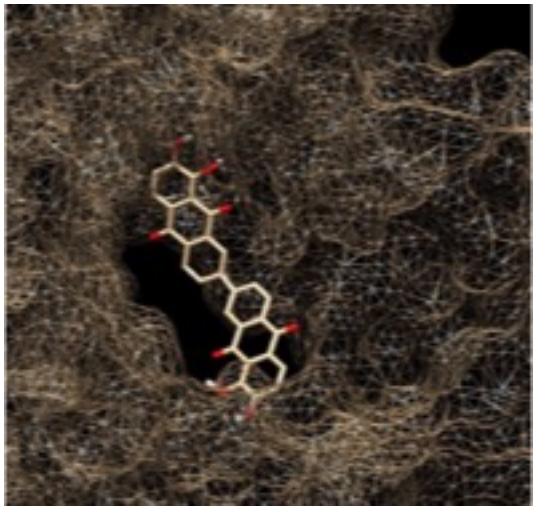


Figure 3. The three-dimensional representation of 7,7'-Bializarin binding to the active site of NSP 12

4 CONCLUSIONS

Using computational programs to find the binding affinities between ligands and the active sites of nonstructural proteins was successful in identifying a large range of ligands that could act as good drugs for SARS-CoV-2. Further research is planned into attempting to identify more ligands that could bind to NSP 12. Additionally, these ligands will be analyzed extensively in Chimera in order to identify the intermolecular forces and other potential causes of good binding affinity. Finally, since these nonstructural proteins are mostly conserved over viruses within the same families, the final drug produced from this research could be extremely effective in treating all SARS viruses, including those that may arise in the future.

5 REFLECTIONS

This project was my first experience with computational research. When I decided to join this project, I was already interested in computational chemistry but did not know the extent of how computational science could bring insights into the mechanics of the real world. At first, the research was intimidating. I was familiar with physical chemistry, but my knowledge of biology and biochemistry was lacking. Additionally, I had not taken any computer science classes, and I was not familiar with the software we used in the research. However, as we began to study the SARS-CoV-2 NSPs, I became fascinated by how different conformations and locations of small molecules could change binding affinities, and therefore have significant impacts on the protein's functionality. Of course, the research was not always easy, and we encountered many obstacles along the way. Still, I am thankful for experience, including the hardships, as I am now much more knowledgeable about proteins, SARS-CoV-2, and generally, how to do research. This project inspired me to pursue studying chemical systems using computational tools at the graduate level.

REFERENCES

[1] Juan Marcelo Carpio Arévalo and Juliana Carolina Amorim. 2021. An in-silico analysis reveals 7,7'-bializarin as a promising DNA gyrase B inhibitor on gram-positive and gram-negative bacteria. *Comput. Biol. Med. 135*, 104626. https://doi.org/10.1016/j.compbiomed.2021.104626

[2] Sargis Dallakyan and Arthur J. Olson. 2015. Small-molecule library screening by docking with PyRx. In *Chemical Biology. Methods in Molecular Biology Vol. 1263*. Humana Press, New York, NY, 243-250. https://doi.org/10.1007/978-1-4939-2269-7_19

[3] Richard T. Eastman, Jacob S. Roth, Kyle R. Brimacombe, Anton Simeonov, Min Shen, Samarjit Patnaik, and Matthew D. Hall. 2020. Remdesivir: A review of its discovery and development leading to emergency use authorization for treatment of COVID-19. *ACS Cent. Sci. 2020*, 6, 672-683. https://doi.org/10.1021/acscentsci.0c00489

[4] Hauke S. Hillen, Goran Kokic, Lucas Farnung, Christian Dienemann, Dimitry Tegunov, and Patrick Cramer. 2020. Structure of replicating SARS-CoV-2 polymerase. *Nature 584*, 154-156. https://doi.org/10.1038/s41586-020-2368-8

[5] John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. 2012. ZINC: A free tool to discover chemistry for biology. *J. Chem. Inf. Model. 52*, 7, 1757-1768. https://doi.org/10.1021/ci3001277

[6] Lindsey S. Jung, Tamara M. Gund, and Mahesh Narayan. 2020. Comparison of binding site of remdesivir and its metabolites with NSP12-NSP7-NSP8, and NSP3 of SARS CoV-2 virus and alternative potential drugs for COVID-19 treatment. *Protein J. 39*, 619-630. https://doi.org/10.1007/s10930-020-09942-9

[7] Robert N. Kirchdoerfer and Andrew B. Ward. 2019. Structure of the SARS-CoV Nsp12 polymerase bound to Nsp7 and Nsp8 co-factors. *Nat. Commun. 10*, 2342. https://doi.org/10.1038/s41467-019-10280-3

[8] Thomas Maier-Komor. n.d. Home of the xjobs Utility. http://www.maier-komor.de/xjobs.html

[9] Giuseppina Mariano, Rebecca J. Farthing, Shamar L. M. Lale-Farjat, and Julien R. C. Bergeron. Structural characterization of SARS-CoV-2: Where we are, and where we need to be. *Front. Mol. Biosci. 7*, 605236. https://doi.org/10.3389/fmolb.2020.605236

[10] Ozal Mutlu, Osman Mutluhan Ugure, Emrah Sariyer, Oguz Ata, Tugba Gul Inci, and Erennur Ugurel. 2022. Targeting SARS-CoV-2 Nsp12/Nsp8 interaction interface with approved and investigational drugs: An *in silico* structure-based approach. *J. Biomol. Struct. Dyn. 40,* 2, 918-930. https://doi.org/10.1080/07391102.2020.1819882

[11] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. UCSF Chimera - A visualization system for exploratory research and analysis. *J. Comput. Chem. 25,* 13, 1605-1612. https://doi.org/10.1002/jcc.20084

[12] Carlos A. Ramos-Guzmán, J. Javier Ruiz-Pernía, and Iñaki Tuñón. Computational simulations on the binding and reactivity of a nitrile inhibitor of the SARS-CoV-2 main protease. *Chem. Commun. 57,* 72, 9096-9099. https://doi.org/10.1039/D1CC03953A

[13] Awadhesh Kumar Singh, Akriti Singh, Ritu Singh, and Anoop Misra. Molnupiravir in COVID-19: A systematic review of literature. *Diabetes Metab. Syndr. Clin. Res. Rev. 15,* 6, 102329. https://doi.org/10.1016/j.dsx.2021.102329

[14] E.J. Snijder, E. Decroly, and J. Ziebuhr. Chapter 3 - The nonstructural proteins directing coronavirus RNA synthesis and processing. In *Advances in Virus Research 96*, 59-126. https://doi.org/10.1016/bs.aivir.2016.08.008

[15] Dave Toth. n.d. Drug_discovery_python_scripts. https://github.com/DaveToth/drug_discovery_python_scripts

[16] Oleg Trott and Arthur J. Olson. 2010. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem. 31*, 2, 455-461. https://doi.org/10.1002/jcc.21334

[17] Philip V'kovski, Annika Kratzel, Silvio Steiner, Hanspeter Stalder, and Volker Thiel. 2021. Coronavirus biology and replication: Implications for SARS-CoV-2. *Nat. Rev. Microbiol. 19,* 3, 155-170. https://doi.org/10.1038/s41579-020-00468-6

[18] Deepa Vasireddy, Rachana Vanaparthy, Gisha Mohan, Srikrishna Varun Malayala, and Paavani Atluri. 2021. Review of COVID-19 variants and COVID-19 vaccine efficacy: What the clinician should know? *J. Clin. Med. Res. 13*, 6, 317-325. https://doi.org/10.14740/jocmr4518

[19] Francis K. Yoshimoto. The proteins of severe acute respiratory syndrome coronavirus-2 (SARS CoV-2 or n-COV19), the cause of COVID-19. *Protein J. 39,* 3, 198-216. https://doi.org/1007/s10930-020-09901-4

[20] ZINC[12]. n.d. Welcome to ZINC - A database of commercially-available compounds. https://zinc12.docking.org

July 2023

Volume 14 Issue 1

ISSN 2153-4136 (online)